
squared

Release 5.2.1

An Pham

May 16, 2024

CONTENTS

| | | |
|-----------|----------------------|------------|
| 1 | Installation | 1 |
| 2 | Browser | 5 |
| 3 | Framework | 7 |
| 4 | Settings | 13 |
| 5 | Methods | 23 |
| 6 | Document | 51 |
| 7 | Release Notes | 91 |
| 8 | References | 113 |
| 9 | Versions | 119 |
| 10 | License | 121 |
| | Index | 123 |

INSTALLATION

- NodeJS 16 (*Minimum*)

1.1 NPX

Listing 1: node_modules

```
npm init
npm i sqd-cli sqd-serve

npx sqd init # --serve-only
npx sqd serve
```

Listing 2: local (dev)

```
npm init
npm i sqd-cli sqd-serve

npx sqd init --public --local-serve
node serve.js
```

Listing 3: without sqd-cli

```
npm init
npm i squared sqd-serve

mkdir dist html
cp -r ./node_modules/squared/dist/* ./dist
cp ./node_modules/squared/html/* ./html      # optional
cp ./node_modules/sqd-serve/config/json/* . # yaml

npx serve
```

1.2 GitHub

Listing 4: prod - master

```
git clone https://github.com/anpham6/squared
cd squared
git checkout master

npm install
npm run build:all

cd ..

git clone https://github.com/anpham6/squared-express
cd squared-express
git checkout master

npm install
npm run prod
npm run deploy # deploy:yaml

cd ../squared

# squared.json
node serve.js
```

Listing 5: dev - 5.2.0

```
git clone https://github.com/anpham6/squared
cd squared
git checkout 5.2.0

npm install
npm run build:dev

cd ..

git clone https://github.com/anpham6/squared-express
```

(continues on next page)

(continued from previous page)

```
cd squared-express
git checkout 3.2.0

npm install
npm run dev
npm run deploy:config # deploy:config:yml

cd ../squared

# squared.json
node serve.js
```

1.3 Repo

Listing 6: Install¹

```
export REPO=$(mktemp /tmp/repo.XXXXXXXXXX)
curl -o ${REPO} https://storage.googleapis.com/git-repo-downloads/repo
gpg --recv-keys 8BB9AD793E8E6153AF0F9A4416530D5E920F5C65
curl -s https://storage.googleapis.com/git-repo-downloads/repo.asc | gpg --verify - $
↪ ${REPO} && install -m 755 ${REPO} ~/bin/repo
```

Listing 7: Usage²

```
mkdir workspace
cd workspace

repo init -u https://github.com/anpham6/squared-repo -m 5.2.0.xml
repo sync
```

Listing 8: ~/workspace

```
android-docs  chrome-docs  e-mc  pi-r  squared  squared-express
```

¹ <https://source.android.com/docs/setup/download#installing-repo>

² <https://source.android.com/docs/setup/reference/repo>

BROWSER

- ES2018

2.1 Download

2.1.1 android

- <https://unpkg.com/squared/dist/squared.min.js>
- <https://unpkg.com/squared/dist/squared.base.min.js>
- <https://unpkg.com/squared/dist/android.framework.min.js>

2.1.2 chrome

- <https://unpkg.com/squared/dist/squared.min.js>
- <https://unpkg.com/squared/dist/squared.base.min.js>
- <https://unpkg.com/squared/dist/chrome.framework.min.js>

2.1.3 vdom

- <https://unpkg.com/squared/dist/squared.min.js>
- <https://unpkg.com/squared/dist/squared.base-dom.min.js>
- <https://unpkg.com/squared/dist/vdom.framework.min.js>

2.1.4 vdom-lite

- <https://unpkg.com/squared/dist/squared.min.js>
- <https://unpkg.com/squared/dist/vdom-lite.framework.min.js>

2.2 Usage

Library files are in the `/dist` folder. A minimum of **two** files are required to run the application.

1. **squared**
2. **squared-base**¹
3. *squared-svg*²
4. **framework**³
5. *extensions*

Usable combinations: 1-2-4 + 1-2-4-5 + 1-2-3-4-5 + 1-vdom-lite

File bundles for common combinations are available in the `/dist/bundles` folder and do not require a call to *set-Framework*.

¹ android | chrome | vdom

² android

³ android | chrome | vdom | vdom-lite

FRAMEWORK

3.1 Android

The primary function `parseDocument()` can be called on multiple elements and multiple times per session. The application will continuously and progressively build the layout files into a single entity with combined shared resources.

Warning: Calling `saveAs()` or `copyTo()` methods before the images or fonts have completely loaded can cause them to be excluded from the generated layout. In these cases you should use the asynchronous `parseDocument()` and not `parseDocumentSync()`.

3.1.1 Example usage

```
<script src="/dist/squared.min.js"></script>
<script src="/dist/squared.base.min.js"></script>
<script src="/dist/squared.svg.min.js"></script>
<script src="/dist/android.framework.min.js"></script>
<script>
  squared.settings.targetAPI = 34; // Optional

  document.addEventListener("DOMContentLoaded", async () => {
    squared.setFramework(android, { /* UserResourceSettingsUI */ });

    await squared.parseDocument(); // Node (document.body)
    /* OR */
    await squared.parseDocument(document.querySelector("main"), "fragment-id", /* ...
↪element */); // Node[]
    /* OR */
    await squared.parseDocument(
      { // Custom settings do not affect other layouts
        element: document.body,
        projectId: "project-1", // Default is "-"

        resourceQualifier: "land", // "res/*" folder suffix
        /* OR */
        resourceQualifier: {
          suffix: "land", // Used for "true" or "undefined" groups
          layout: true, // Will copy to "res/layout-land" when "suffix" is defined
          font: false, // Will not copy anything to "res/font" or "res/font-land"
        }
      }
    );
  });
}
```

(continues on next page)

(continued from previous page)

```

    image: "hdpi", // Will copy to "res/drawable-hdpi"
    video: "w720dp", // Will copy to "res/raw-w720dp"
    audio: "w720dp", // Same as "video" and treated separately
    animation: "v34", // Will copy to "res/anim-v34"
    menu: "", // Will copy to default location "res/menu"
    string: undefined, // Will copy to default location "res/values" when "suffix"
    ↪is undefined
    integer: undefined,
    fraction: undefined,
    array: undefined,
    color: undefined,
    dimension: undefined,
    style: undefined,
    theme: undefined
  },
  excludeQuery: [
    { selector: "main > article" }, // Hide elements
    {
      // Bypass processing routines
      resource: NODE_RESOURCE.BOX_STYLE, // squared.base.lib.constant
      procedure: NODE_PROCEDURE.OPTIMIZATION,
      section: APP_SECTION.DOM_TRAVERSE
    }
  ],

  /* Extensions */
  include: ["android.substitute"], // Automatically removed after finalize
  exclude: ["squared.list", "squared.grid"], // Disabled only during parseDocument

  /* Optional features */
  enabledMultiline: false,
  enabledSubstitute: true,

  /* Callback events */
  afterCascade(sessionId, node) { /* Restore previous state */},
  beforeCascade(sessionId) { /* Edit element attributes */},
  beforeRender(layout) { /* Edit initial values */},
  afterFinalize(node) { /* Edit controller values */}
},
{ // Only "element" is required
  element: "fragment-1",
  projectId: "project-1", // Implicit once projectId is not "-"
  resourceQualifier: "port",
  pathname: "app/src/main/res/layout-hdpi", // Will not be overridden by
  ↪resourceQualifier "port"
  filename: "fragment.xml",
  baseLayoutAsFragment: ["com.example.fragment", "fragment-tag", "fragment_id"]
}
);
await squared.parseDocument({
  element: "fragment-2",
  projectId: "project-2", // Will not conflict with projectId "project-1"

```

(continues on next page)

(continued from previous page)

```

    resourceQualifier: "port",
    enabledFragment: true,
    fragmentableElements: [
      { selector: "main", name: "com.example.fragment", filename: "fragment.xml",
↪documentId: "main_content" }, // document.querySelector
      "main > article" // document.querySelectorAll (declarative double nested
↪fragments are invalid)
    ],
    options: {
      "android.resource.fragment": {
        dynamicNestedFragments: true // FragmentContainerView or FrameLayout as the
↪container (name and tag are ignored)
      }
    }
  });

  await squared.close("project-1"); // Next call to "parseDocument" will reset project
↪(optional)

  // File actions - implicitly calls "close"

  await squared.save(); // Uses defaults from settings
  /* OR */
  await squared.saveAs("project.zip", { /* RequestData */ });
  await squared.saveAs("default.7z", { throwErrors: true }).catch(err => console.
↪log(err)); // Will cancel partial archive download
  /* OR */
  await squared.copyTo("/path/project-1", { projectId: "project-1", emptyDir: true,
↪ignoreExtensions: true });
  /* OR */
  await squared.appendTo("http://localhost:3000/archives/project.001", { format: "7z" }
↪);

  squared.reset(); // Start new "parseDocument" session (optional)
});
</script>

```

Listing 1: Cross-origin support

```

squared.prefetch("css").then(() => squared.parseDocument()); // Chromium
/* OR */
Promise.all(
  squared.prefetch("css", true), // All stylesheets
  squared.prefetch("css", "./undetected.css", element.shadowRoot),
  squared.prefetch("svg", "http://embedded.example.com/icon.svg", "../images/android.svg
↪")
)
.then(() => squared.parseDocument());

```

Listing 2: Kill request

```
squared.kill("30s").then(pid => { // Abort next request in 30 seconds
  if (pid > 0) {
    /* KILLED */
  }
});
/* OR */
await squared.saveAs("project.zip", { timeout: 10 }); // Cancels request if not complete,
↪ in 10 seconds
```

Listing 3: Modify attributes

```
squared.parseDocument().then(() => {
  const body = squared.findDocumentNode(document.body);
  body.android("layout_width", "match_parent");
  body.lockAttr("android", "layout_width");
});
```

Listing 4: Observe element attributes

```
await squared.parseDocument({
  element: document.body,
  observe(mutations, observer, settings) {
    squared.reset(); // Required after a File action
    squared.parseDocument(settings).then(() => {
      squared.copyTo("/path/project", { modified: true }).then(response => {
        console.log(response);
      });
    });
  }
});
squared.observe();
```

Listing 5: Observe element source files

```

await squared.observeSrc(
  "link[rel=stylesheet]",
  (ev, element) => {
    squared.reset();
    squared.parseDocument().then(() => squared.copyTo("/path/project"));
  },
  { // squared.json
    port: 8080,
    secure: false,
    action: "reload",
    expires: "1h"
  }
);

```

3.2 Chrome

Uses the *VDOM* framework as the element base but can also export any used resources for further processing. It is adequate for most single page applications and gives you the ability to preview your application before building a working copy of it.

3.2.1 Example usage

```

<script src="/dist/squared.min.js"></script>
<script src="/dist/squared.base.min.js"></script>
<script src="/dist/chrome.framework.min.js"></script>
<script>
  document.addEventListener("DOMContentLoaded", async () => {
    squared.setFramework(chrome, { /* UserResourceSettings */ });

    await squared.save(); // Uses defaults from settings
    /* OR */
    await squared.saveAs("android.zip", { /* RequestData */ });
    /* OR */
    await squared.copyTo("/path/project", { /* RequestData */ });
    /* OR */
    await squared.appendTo("/path/to/android.7z", { /* RequestData */ });
  });
</script>

```

Listing 6: Observe element

```

await squared.copyTo("/path/project", { useOriginalHtmlPage: false, observe: true }).
  then(() => {
    squared.observe(); // Watch all events
    /* OR */
    squared.observe({
      subtree: true,

```

(continues on next page)

(continued from previous page)

```
    childList: true,
    attributes: true,
    characterData: true,
    attributeOldValue: true,
    characterDataOldValue: true
  });
});
```

3.3 VDOM

A minimal framework that is primarily for debugging elements in [DevTools](#) console. The *lite* version is half the download size (*55kb gzipped*) and was built for browser applications.

3.3.1 Example usage

Listing 7: base-dom

```
<script src="/dist/squared.min.js"></script>
<script src="/dist/squared.base-dom.min.js"></script>
<script src="/dist/vdom.framework.min.js"></script>
<script>
  document.addEventListener("DOMContentLoaded", async () => {
    squared.setFramework(vdom, { /* UserSettings */ });

    const elements = await squared.querySelectorAll("");
    /* OR */
    const element = await squared.fromElement(document.body);
    /* OR */
    const elements = await squared.getElementById("content-id").querySelectorAll("*");
  });
</script>
```

Listing 8: lite

```
<script src="/dist/squared.min.js"></script>
<script src="/dist/vdom-lite.framework.min.js"></script>
<script>
  document.addEventListener("DOMContentLoaded", () => {
    squared.setFramework(vdom);

    const element = squared.querySelector("body", true); // Synchronous
    /* OR */
    const element = squared.fromElement(document.body, true);
  });
</script>
```

Tip: Synchronous is the recommended way to query when you are not concerned with image dimensions.

SETTINGS

4.1 Interface

These settings are available in the global variable `squared` to customize your desired output structure. Each framework shares a common set of settings and also a subset of their own settings.

```
interface UserSettings {  
  builtInExtensions: string[];  
  showErrorMessage: boolean;  
  pierceShadowRoot: boolean;  
  adaptStyleMap: boolean;  
  createElementMap: boolean;  
}
```

```
interface UserResourceSettings extends UserSettings {  
  preloadImages: boolean;  
  preloadFonts: boolean;  
  preloadLocalFonts: boolean;  
  preloadCustomElements: boolean;  
  outputEmptyCopyDirectory: boolean;  
  outputDocumentHandler: string | string[];  
  outputArchiveFormat: string;  
  outputArchiveName: string;  
  outputArchiveCache: boolean;  
  outputSummaryModal: boolean | string;  
  formatUUID?: string;  
  formatDictionary?: string;  
  outputTasks?: Record<string, TaskCommand[]>;  
  outputWatch?: Record<string, boolean | WatchInterval>;  
  outputConfigName?: string;  
  observePort?: number;  
  observeSecurePort?: number;  
  observeExpires?: string;  
  broadcastPort?: number;  
  broadcastSecurePort?: number;  
}
```

```
interface UserResourceSettingsUI extends UserResourceSettings {  
  enabledSVG: boolean;  
  supportNegativeLeftTop: boolean;  
}
```

(continues on next page)

(continued from previous page)

```

    showAttributes: boolean | Record<string, unknown>;
    showComments: boolean | CssStyleAttr[] | { include?: ShowCommentsInclude } & Record
    <string, CssStyleAttr[]>;
    insertSpaces: number;
    outputDirectory: string;
    outputDocumentEditing: boolean;
    outputDocumentCSS: CssStyleAttr[];
    framesPerSecond?: number;
    useShapeGeometryBox?: boolean;
    resolutionScreenWidth?: number;
    resolutionScreenHeight?: number;
}

```

Added in version 5.2.0:

- *UserSettings* property **adaptStyleMap** for the rendered value was created.
- *UserResourceSettingsUI* property **useShapeGeometryBox** for SVG native method `getBbox` was created.
- *UserResourceSettingsUI* property **showAttributes** for global attribute modification is supported.

4.1.1 Example usage

Listing 1: All

```

squared.settings = {
  pierceShadowRoot: false,
  adaptStyleMap: false, // Merge all associated styles into rendered value
  showErrorMessage: false, // console.log
  showErrorMessage: true, // alert
  createElementMap: false // Cache queries for subsequent nested queries
};

```

Note: The native `document.querySelector` does not enter *ShadowRoot* elements.

Listing 2: Resource

```

squared.settings = {
  formatUUID: "8-4-4-4-12", // UUID: 8-4-[12345]3-[89ab]3-12
  formatDictionary: "0123456789abcdef",
  outputConfigName: "sqd.config", // Per directory filename with URL globs of static
  <pages
  outputTasks: {
    "*.xml": { handler: "gulp", task: "minify" }
  },
  outputWatch: {
    "**/images/*.png": true,
    "**/images/*.jpg": { interval: 1000, expires: "2h" }
  },
  observePort: 8080,
  observeSecurePort: 8443,

```

(continues on next page)

(continued from previous page)

```
observeExpires: "1h", // Server defaults will be used
broadcastPort: 3080,
broadcastSecurePort: 3443
};
```

Attention: These settings are not available in the *VDOM* framework.

4.2 Android

4.2.1 Interface

```
interface UserResourceSettingsUI {
    targetAPI: number;
    supportRTL: boolean;
    compressImages: boolean;
    enabledMultiline: boolean;
    enabledViewModel: boolean;
    enabledIncludes: boolean;
    enabledFragment: boolean;
    enabledSubstitute: boolean;
    enabledCompose: boolean;
    dataBindableElements: ExtensionViewModelElement[];
    includableElements: ExtensionIncludeElement[];
    substitutableElements: ExtensionSubstituteElement[];
    fragmentableElements: (string | ExtensionFragmentElement)[];
    composableElements: string[];
    baseLayoutAsFragment: boolean | string | string[] | ExtensionFragmentElement;
    convertPixels: ResolutionUnit;
    lineHeightAdjust: number;
    fontMeasureAdjust: number;
    lockElementSettings: boolean;
    customizationsBaseAPI: number | number[];
    customizationsOverwritePrivilege: boolean;
    removeDeprecatedAttributes: boolean | string[];
    removeUnusedResourceViewId: boolean;
    preferMaterialDesign: boolean | "MaterialComponents" | "Material3";
    idNamingStyle: NamingStyles;
    showAttributes: boolean | Record<string, (string | null)[] | string | null>;
    createDownloadableFonts: boolean;
    manifestActivityName: string;
    manifestLabelAppName: string;
    manifestThemeName: string;
    manifestParentThemeName: string;
    createManifest: boolean;
    createBuildDependencies: boolean | ("ktx" | "baseline-profile")[];
    outputMainFileName: string;
    outputFragmentFileName: string;
    resolutionDPI?: number;
```

(continues on next page)

(continued from previous page)

```

resourceQualifier?: string | ResourceQualifierMap;
resourceSystemColors?: Record<string, string | [string, number] | ColorRGB>;
convertImages?: string;
convertLineHeight?: ResolutionUnit;
baseLayoutToolsIgnore?: string;
manifestPackage?: string;
}

```

Added in version 5.2.0:

- *UserResourceSettingsUI* property **showAttributes** accepts a map to globally replace layout attributes.
- *UserResourceSettingsUI* property **resourceSystemColors** device color translation map was created.
- *ExtensionFragmentElement* extends the *ViewAttribute* interface.

4.2.2 Example usage

Listing 3: Customizable (project/all)

```

squared.settings = {
  targetAPI: 34,
  resolutionDPI: 160, // 320dpi = 2560x1600
  resolutionScreenWidth: 1280,
  resolutionScreenHeight: 800,
  framesPerSecond: 60, // SVG animation only
  useShapeGeometryBox: true, // Dimensions use native SVG method getBbox
  supportRTL: true,
  supportNegativeLeftTop: true,
  preloadImages: true,
  preloadFonts: true,
  preloadLocalFonts: true, // Chromium
  preloadCustomElements: true, // pierceShadowRoot = true
  enabledSVG: true, // android.resource.svg
  enabledMultiline: true, // android.delegate.multiline
  enabledViewModel: true, // android.resource.data
  enabledIncludes: false, // android.resource.includes
  enabledSubstitute: false, // android.resource.fragment
  enabledFragment: false, // android.substitute
  enabledCompose: false, // android.compose.view
  dataBindableElements: [/* ExtensionViewModelElement */],
  includableElements: [/* ExtensionIncludeElement */],
  substitutableElements: [/* ExtensionSubstituteElement */],
  fragmentableElements: [/* "selector" | ExtensionFragmentElement */],
  composableElements: [/* "selector" | "--property" */],
  baseLayoutAsFragment: "fragment-name",
  baseLayoutAsFragment: ["fragment-name", "fragment-tag", "document_id" /* Optional */],
  baseLayoutAsFragment: { // ExtensionFragmentElement
    name: "androidx.navigation.fragment.NavHostFragment",
    documentId: "main_content",
    app: {
      navGraph: "@navigation/product_list_graph",

```

(continues on next page)

(continued from previous page)

```

    defaultNavHost: "true"
  }
},
baseLayoutToolsIgnore: "TooManyViews, HardcodedText", // Android Studio Editor
fontMeasureAdjust: 0.75, // thicker < 0 | thinner > 0
lineHeightAdjust: 1.1, // shorter < 1 | taller > 1
preferMaterialDesign: true, // "Material3"
preferMaterialDesign: "MaterialComponents",
createDownloadableFonts: true,
createElementMap: false, // Cache not used with NodeUI
pierceShadowRoot: true,
adaptStyleMap: true, // Use rendered values for output
lockElementSettings: false, // Modify Node before rendering (LocalSettingsUI)
customizationsBaseAPI: -1, // None
customizationsBaseAPI: 0, // All (14 - 34)
customizationsBaseAPI: [0, 33, 34], // Multiple with ordering
customizationsOverwritePrivilege: true, // Existing auto-generated attributes (e.g.
↳ layout_width)
removeDeprecatedAttributes: true, // Remove all
removeDeprecatedAttributes: ["enabled", "singleLine"], // Remove all except the listed
↳ values
removeUnusedResourceViewId: false,
idNamingStyle: "android", // Use layout name
idNamingStyle: "html", // Use tagName
idNamingStyle: {
  "__default__": "html", // Optional
  "DIV": "comments", // HTML is uppercase (comments_1 then comments_2)
  "svg": ["vector", 0], // SVG is lowercase (vector_0 then vector_1)
  "#text": "text", // Plain text
  "::first-letter": "dropcap", // Pseudo element
  "main > section": ["content", 1, 2], // content_1 then content_3
  "form input[type=submit]": function(node) {
    return "submit_" + node.id;
  }
},
outputMainFileName: "activity_main.xml",
outputFragmentFileName: "fragment_main.xml"
};

```

Listing 4: Customizable (project/main)

```

squared.settings = {
  resourceQualifier: "land", // "res/layout-land"
  resourceSystemColors: {
    "system_accent1_100": "white", // Will be converted to ARGB
    "system_accent1_200": ["#ff0000", 0.75], // opacity
    "system_accent1_300": squared.lib.color.parseColor("#000", 1)
  },
  manifestPackage: "example", // <manifest package="example"> (OR: RequestData{
↳ namespace: "android.application.id" }>)
  manifestLabelAppName: "android", // <application android:label="@string/android">
  manifestThemeName: "AppTheme", // <application android:theme="@style/AppTheme">

```

(continues on next page)

(continued from previous page)

```

↳ (overrides manifestParentThemeName)
    manifestParentThemeName: "Theme.AppCompat.Light.NoActionBar", // <style parent="Theme.
↳ AppCompat.Light.NoActionBar"> [res/values/styles.xml]
    manifestActivityName: ".MainActivity", // <activity android:name=".MainActivity">
    outputDocumentEditing: true, // RequestData{ targetAPI + dependencies + mainParentDir,
↳ + mainSrcDir + directories + dataBinding + elements }> (append without overwrite)
    outputDocumentCSS: [], // CSS properties to be processed with a server extension (e.g.
↳ "boxShadow")
    outputDirectory: "app/src/main",
    createManifest: false, // Update AndroidManifest.xml
    createBuildDependencies: false, // build.gradle
    createBuildDependencies: ["ktx", "baseline-profile"]
};

```

Listing 5: Global

```

squared.settings = {
    builtInExtensions: [
        "squared.accessibility",
        "android.delegate.background",
        "android.delegate.negative-x",
        "android.delegate.positive-x",
        "android.delegate.max-width-height",
        "android.delegate.percent",
        "android.delegate.content",
        "android.delegate.scrollbar",
        "android.delegate.radiogroup",
        "android.delegate.multiline",
        "squared.relative",
        "squared.css-grid",
        "squared.flexbox",
        "squared.table",
        "squared.column",
        "squared.list",
        "squared.grid",
        "squared.sprite",
        "squared.whitespace",
        "android.resource.background",
        "android.resource.svg",
        "android.resource.strings",
        "android.resource.fonts",
        "android.resource.dimens",
        "android.resource.styles",
        "android.resource.data"
    ],
    convertImages: "png", // jpeg | webp | gif | bmp
    compressImages: false, // TinyPNG (https://tinypng.com/developers)
    showAttributes: {
        "android:hyphenationFrequency": "full", // Replace all
        "android:fontFeatureSettings": null // Delete all
    },
    showComments: false, // <!-- TODO in layout.xml -->
}

```

(continues on next page)

(continued from previous page)

```

showComments: ["boxShadow"],
showComments: {
    self: ["boxShadow"],
    nextSibling: ["marginBottom"],
    previousSibling: ["marginTop"],
    parent: ["position", "top", "left"]
},
showComments: {
    self: ["boxShadow", ".className"],
    include: {
        tagName: true, // ["button"]
        attributes: true, // ["id", "style"]
        dataset: false,
        bounds: true
    }
},
showErrorMessages: false,
convertPixels: "dp", // ResolutionUnit
convertLineHeight: "sp", // ResolutionUnit
insertSpaces: 0, // tabs
insertSpaces: 4, // per tab
outputDocumentHandler: "android",
outputEmptyCopyDirectory: false, // Sub directories within target directory (OR:
↳RequestData<{ emptyDir: false }>)
outputSummaryModal: true, // Affected files in base output directory
outputSummaryModal: "path/summary.css", // Use custom style sheet
outputSummaryModal: ".status-4 { color: purple; }", // Use inline style sheet
outputTasks: {
    "**/drawable/*.xml": { handler: "gulp", task: "minify" }
},
outputWatch: {
    "**/drawable/*.png": true,
    "**/drawable/*.jpg": { interval: 1000, expires: "2h" }
},
outputArchiveName: "android-xml", // squared.saveAs
outputArchiveFormat: "zip", // tar | 7z | gz
outputArchiveCache: false // Downloadable URL in ResponseData<downloadUrl>
};

```

Listing 6: Global (optional)

```
squared.settings = {  
  builtInExtensions: [  
    "android.resource.includes", // enabledIncludes  
    "android.substitute", // enabledSubstitute  
    "android.resource.fragment", // enabledFragment  
    "jetpack.compose.view" // enabledCompose  
  ]  
};
```

4.3 Chrome

The default settings were optimized for creating static web pages with [E-mc](#). Any customizations are usually the output properties.

4.3.1 Interface

```
interface UserResourceSettings {  
  excludePlainText?: boolean;  
  websocketPort?: number;  
  websocketSecurePort?: number;  
}
```

4.3.2 Example usage

Listing 7: Global (bundle)

```
squared.settings = {  
  pierceShadowRoot: true,  
  adaptStyleMap: false,  
  builtInExtensions: [], // Called for every local asset (ext.processFile)  
  showErrorMessage: false,  
  websocketPort: 80, // Used with watch  
  websocketSecurePort: 443,  
  outputDocumentHandler: "chrome",  
  /* Same as android */  
  outputEmptyCopyDirectory: false,  
  outputSummaryModal: false,  
  outputTasks: {  
    "*.js": [{ handler: "gulp", task: "minify" }, { handler: "gulp", task: "beautify" }]  
  },  
  outputWatch: { "*": true },  
  outputArchiveName: "chrome-data",  
  outputArchiveFormat: "zip",  
  outputArchiveCache: false  
};
```


Listing 8: Global (query)

```
squared.settings = {  
  excludePlainText: true, // HTMLElement + SVGElement  
  createElementMap: true  
};
```

Listing 9: Global (unused)

```
squared.settings = {  
  preloadImages: false,  
  preloadFonts: false,  
  preloadLocalFonts: false,  
  preloadCustomElements: false  
};
```

4.4 VDOM

4.4.1 Interface

```
import type { UserSettings } from "../interface";
```

4.4.2 Example usage

Listing 10: Global

```
squared.settings = {  
  createElementMap: true,  
  pierceShadowRoot: false,  
  adaptStyleMap: false,  
  showErrorMessages: false  
};
```

Listing 11: Global (unused)

```
squared.settings = {  
  builtInExtensions: []  
};
```

Tip: Frequent modifications to the DOM use the setting `createElementMap = false`.

4.5 Local Storage

Custom named user settings per framework can be saved to [local storage](#) as JSON and reused across all pages in the same domain. Extensions are configured using the same procedure.

4.5.1 Framework

Listing 12: Save

```
squared.setFramework(android, { compressImages: true }, "android-example");
```

Listing 13: Load

```
squared.setFramework(android, "android-example");
```

4.5.2 Output

Listing 14: Save

```
await squared.copyTo("/path/project", { /* options */ }, "copy-example", true); // Will  
↪ overwrite when "true" and not merge with previously saved settings
```

Listing 15: Load

```
// Local storage  
await squared.copyTo("/path/project", { /* options */ }, "copy-example"); // Object.assign(  
↪ { ...copy-example }, options)  
await squared.copyTo("/path/project", "copy-example"); // options = { ...copy_example }  
  
// External  
await squared.copyTo("/path/project", { /* options */ }, "http://localhost:3000/copy-to/  
↪ base-config.json"); // Object.assign({ ...base-config }, options)  
await squared.copyTo("/path/project", "http://localhost:3000/copy-to/base-config.json");  
↪ // options = { ...base-config }
```

METHODS

5.1 Interface

5.1.1 squared

Listing 1: Serve (API)

```
function setHostname(value: string): void;
function setEndpoint(name: string, value: string): void;
function setLocalAddress(...values: (string | URL)[]): void;
function auth(token: string): void;

function kill(timeout: string): Promise<number>;
function kill(pid?: number, timeout?: number | string): Promise<number>;

function broadcast(callback: BroadcastMessageCallback, socketId: string): boolean;
function broadcast(callback: BroadcastMessageCallback, options: FileBroadcastOptions): ↵
↵boolean;
```

Listing 2: Framework

```
function setFramework(value: Application, options?: PlainObject, cache?: boolean): void;
function setFramework(value: Application, loadAs: string, cache?: boolean): void;
function setFramework(value: Application, options?: PlainObject, saveAs?: string, cache?
↵: boolean): void;

function extend(functionMap: PlainObject, framework?: number): void;
function clear(): void;
```

Listing 3: Extensions

```
function add(...values: ExtensionRequestObject[]): number;
function remove(...values: ExtensionRequest[]): number;
function get(...values: string[]): Extension | Extension[] | undefined;
function attr(name: ExtensionRequest, attrName: string, value?: unknown): unknown;

function apply(value: ExtensionRequest, saveAs: string): boolean;
function apply(value: ExtensionRequest, options: PlainObject, saveAs?: string): boolean;
```

Listing 4: Session (*create*)

```
function prefetch(type: "css", all: boolean, ...targets: (Document | ShadowRoot)[]): Promise<PrefetchItem[]>;
function prefetch(type: "css", ...targets: (Document | ShadowRoot)[]): Promise<PrefetchItem[]>;
function prefetch(type: "javascript" | "image" | "svg", all: boolean, ...targets: string[]): Promise<PrefetchItem[]>;
function prefetch(type: "javascript" | "image" | "svg", ...targets: string[]): Promise<PrefetchItem[]>;

function parseDocument(...elements: RootElement[]): Promise<Node | Node[] | void>;
function parseDocumentSync(...elements: RootElement[]): Node | Node[] | undefined;
```

Listing 5: Session (*modify*)

```
function findDocumentNode(value: HTMLElement | string, projectId?: string): Node | undefined;
function findDocumentNode(value: HTMLElement | string, all: true, projectId?: string): Node[];

function latest(value = 1): string;
function latest(value: 1 | -1): string;
function latest(value: number): string[];

function close(projectId?: string): Promise<boolean>;
function reset(projectId?: string): void;
```

Listing 6: Session (*API*)

```
function save(): FileActionResult;
function save(timeout: number): FileActionResult;
function save(projectId: string, timeout: number): FileActionResult;
function save(projectId: string, broadcastId?: string): FileActionResult;

function saveAs(value: string, setting: string): FileActionResult;
function saveAs(value: string, options?: FileActionOptions, setting?: string, overwrite?: boolean): FileActionResult;

function appendTo(value: string, setting: string): FileActionResult;
function appendTo(value: string, options?: FileActionOptions, setting?: string, overwrite?: boolean): FileActionResult;

function copyTo(value: string | string[], setting: string): FileActionResult;
function copyTo(value: string | string[], options?: FileActionOptions, setting?: string, overwrite?: boolean): FileActionResult;

function saveFiles(value: string, setting: string): FileActionResult;
function saveFiles(value: string, options: FileActionOptions, setting?: string, overwrite?: boolean): FileActionResult;

function appendFiles(value: string, setting: string): FileActionResult;
```

(continues on next page)

(continued from previous page)

```
function appendFiles(value: string, options: FileActionOptions, setting?: string,
↳ overwrite?: boolean): FileActionResult;

function copyFiles(value: string | string[], setting: string): FileActionResult;
function copyFiles(value: string | string[], options: FileActionOptions, setting?:
↳ string, overwrite?: boolean): FileActionResult;
```

Listing 7: DOM

```
function getElementById(value: string, sync: true, cache = true): Node | null;
function getElementById(value: string, sync?: false, cache = true): Promise<Node | null>;

function querySelector(value: string, sync: true, cache = true): Node | null;
function querySelector(value: string, sync?: false, cache = true): Promise<Node | null>;

function querySelectorAll(value: string, sync: true, cache = true): Node[];
function querySelectorAll(value: string, sync?: false, cache = true): Promise<Node[] |
↳ null>;

function fromElement(element: HTMLElement | string, sync: true, cache?: boolean): Node |
↳ null;
function fromElement(element: HTMLElement | string, sync?: false, cache?: boolean):
↳ Promise<Node | null>;

function fromNode(node: Node, sync: true, cache?: boolean): Node | null;
function fromNode(node: Node, sync?: false, cache?: boolean): Promise<Node | null>;
```

Listing 8: Observe

```
function observe(enable = true): void;
function observe(options: MutationObserverInit): void;

function observeSrc(element: HTMLElement | string, options: FileObserveOptions): Promise
↳ <ObserveSocket | ObserveSocket[]>;
function observeSrc(element: HTMLElement | string, callback: (ev: MessageEvent, target:
↳ HTMLElement) => void, options?: FileObserveOptions): Promise<ObserveSocket |
↳ ObserveSocket[]>;
```

5.1.2 android

```
interface AppFramework {
  setViewModel(data: AppViewModel, sessionId?: string): void;
  setViewModelByProject(data: AppViewModel, projectId?: string): void;
  addDependency(group: string, name: string, version?: number | string, type?: number
↳ | boolean, overwrite?: boolean): string;
  addDependencyByProject(projectId: string, group: string, name: string, version?:
↳ number | string, type?: number | boolean, overwrite?: boolean): string;
  addFontProvider(authority: string, package: string, certs: string[], webFonts:
↳ string | FontProviderFonts): boolean | Promise<boolean>;
  addXmlNs(name: string, uri: string): void;
```

(continues on next page)

(continued from previous page)

```

    customize(api: number, widget: string, options: Record<string, StringMap>): Record
    <string, StringMap> | undefined;
    loadCustomizations(name: string): void;
    saveCustomizations(name: string): void;
    resetCustomizations(): void;
    setResolutionByDeviceName(value: string): boolean;
    getLocalSettings(): ControllerSettingsUI;
    removeObserver(element: HTMLElement): boolean;
}

```

5.1.3 chrome

```

interface AppFramework {
    removeObserver(element: HTMLElement): boolean;
}

```

5.2 squared

setHostname(value)

Use another CORS (Cross-Origin Resource Sharing)-enabled server for processing assets.

Arguments

- **value** (string()) – http(s)://hostname(:port)

Usage:

```

squared.setHostname("https://localhost:8000");

squared.setHostname(); // Reset to window.location (e.g. localhost:3000)

```

setEndpoint(name, value)

Set alternate pathname for API v1 functions.

Arguments

- **name** (string()) – *ASSETS_COPY* | *ASSETS_ARCHIVE* | *LOADER_DATA* | *THREADS_KILL* | *WEBSOCKET_OBSERVE*
- **value** (string()) – Absolute path to server *GET/POST* method

Usage:

```

squared.setEndpoint("ASSETS_COPY", "/api/v2/assets/copy");

```

setLocalAddress(...values)

Additional hostnames which are interpreted as localhost. Protocol and port are not required.

Arguments

- **values** (string()) – Same format as URL.hostname

Usage:

```
squared.setLocalAddress("127.0.0.1", "nodejs-001");
```

Alternate:

```
squared.setLocalAddress(new URL("http://0.0.0.0"));
```

auth(*token*)

Set JWT (JSON Web Token) authorization token for all requests

Arguments

- **token** (string()) – Three concatenated Base64url-encoded strings separated by dots

Usage:

```
squared.auth("eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
↪eyJ1c2VybmFtZSI6InNxdWFyZWQiLCJwYXNzd29yZCI6Imp3dDEyMyIsIm1hdCI6MTY1MjcxNDQwMH0.
↪xqwQ96LsItilsB1dskzJQqXORaZ4mGMu0FeZezo297c");
```

kill([*pid*, *timeout*])

Abort request if not completed in the given amount of time.

Arguments

- **pid** (number()) – (optional) File request self-assigned number (*options.pid*)
- **timeout** (number()) – (optional) Seconds until request expires

Returns

Promise<number>

Usage:

```
squared.saveAs("example.zip", { pid: 100 });
squared.kill(100).then(result => {
  if (result > 0) {
    /* KILLED */
  }
});

squared.kill(); // Terminate previous request
squared.kill(0); // By username (auth required)

squared.kill(-1, 10); // Terminate previous request in 10 seconds
squared.kill(NaN, 0.5); // Terminate next request in 500 milliseconds
```

Alternate:

```
squared.kill(-1, "10s"); // Terminate previous request in 10 seconds
squared.kill("500ms"); // Terminate next request in 500 milliseconds
```

broadcast(*callback*, *socketId*)

Redirect stdout messages to DevTools console.

Arguments

- **callback** (function()) – See [BroadcastMessageCallback](#)
- **socketId** (string()) – Unique identifier assigned during server initialization

Returns

boolean

Usage:

```
squared.broadcast(result => { console.log(result.value); }, "111-111-111"); // Uses  
↪ "socketId"  
squared.broadcast(result => { console.log(result.value); }, "222-222-222");  
  
squared.copyTo("/path/to/project", { broadcastId: ["111-111-111", "222-222-222"] });
```

Alternate:

```
squared.copyTo("/path/to/project", {  
  broadcast: {  
    socketId: "111-111-111",  
    callback: result => {  
      console.log(result.value);  
    }  
  }  
});  
  
// Messages sent from another API (FileBroadcastOptions)  
squared.broadcast(result => { console.log(result.value); }, { // Uses "socket_id"  
  socketId: "333-333-333",  
  socketKey: "socket_id",  
  secure: true,  
  hostname: "1.1.1.1",  
  port: 8080  
});
```

setFramework(target[, options, cache])Install application interpreter. (e.g. *android.framework.js*)**Arguments**

- **target** (object()) – Global object implementing *AppFramework*
- **options** (object()) – (optional) Initialize settings with non-default values
- **cache** (boolean()) – (optional) Load previous cached instance and settings

Usage:

```
squared.setFramework(android);  
squared.setFramework(android, true); // Used when switching frameworks  
squared.setFramework(android, { targetAPI: 34, idNamingStyle: "html" });
```

Alternate:

```
// Save  
squared.setFramework(android, { targetAPI: 26, enabledIncludes: true }, "example");  
↪ // Local storage  
  
// Load  
squared.setFramework(android, "example");
```


extend(*map*[, *framework*])

Add functions and initial variables to the *Node* prototype including overwriting preexisting class definitions. Accessor properties are supported using the *get/set* object syntax.

Arguments

- **map** (object()) – Attribute object consisting of extensions and overrides
- **framework** (number()) – (optional) See *APP_FRAMEWORK*

Usage:

```
squared.extend({
  _id: 1,
  altId: {
    get() {
      return this._id;
    },
    set(value) {
      this._id += value;
    }
  },
  customId: {
    value: 2,
    configurable: false,
    enumerable: false
  },
  addEvent(eventName, callback) {
    this.element.addEventListener(eventName, callback);
  }
});
squared.setFramework(vdom);

const body = await squared.fromElement(document.body);
body.altId = 2; // 3
body.altId = 2; // 5
body.addEvent("click", function (ev) {
  this.classList.toggle("example");
});

squared.extend({ _preferInitial: true }, APP_FRAMEWORK.VDOM /* 1 */ | APP_FRAMEWORK.
↳CHROME /* 4 */); // squared.base.lib.constant
squared.setFramework(chrome);
```

clear()

Calls *Application.clear()* for any loaded frameworks and deletes all cached sessions.

Usage:

```
squared.clear();
```

add(...targets)

Include extensions to be processed with any built-in extensions.

Arguments

- **targets** – Name of disabled extension or control implementing *Extension*

- **targets** – Tuple of extension and configuration object

Returns

number

Usage:

```
class Drawer extends squared.base.ExtensionUI {
  constructor(name, framework, options) {
    super(name, framework, options);
    this.options = {
      element: {},
      resource: {},
      self: {},
      navigationView: {}
    };
    this.documentBase = true;
    this.require({ name: "android.external", leading: true });
    this.require("android.widget.menu");
    this.require("android.widget.coordinator");
  }
}

const drawer = new Drawer("android.widget.drawer", squared.base.lib.constant.APP_
↳FRAMEWORK.ANDROID /* 2 */);
squared.add(drawer);

// Built-in
squared.add("android.resource.includes");

const options = {
  element: {
    content: { android: { layout_width: "match_parent" } }
  }
};
squared.add(["android.substitute", options]);
```

remove(...targets)

Exclude extensions by name or control.

Arguments

- **targets** – Name or control of extensions

Returns

number

Usage:

```
const drawer = new Drawer("android.widget.drawer", 2);
squared.add(drawer);

squared.remove(drawer);
/* OR */
squared.remove("android.widget.drawer");
```

get(...targets)

Retrieve extensions by name only.

Arguments

- **targets** (string()) – Name of extension

Returns

- *Extension* | undefined
- *Extension[]* (multiple)

Usage:

```
const drawer = squared.get("android.widget.drawer");
const [drawer, menu] = squared.get("android.widget.drawer", "android.widget.menu");

drawer.options.navigationView.android = { fitsSystemWindows: "true" };
menu.project.set(document.getElementById("child-item-id"), await fetch("http://
↪localhost:3000/drawer/menu.json"), "project-1" /* optional */); // Add project
↪data
```

attr(target, name[, value])

Set or get extension options. **typeof** is enforced and will only set existing attributes.

Arguments

- **target** – Name or control of extension
- **name** (string()) – Name of attribute in *Extension*.options
- **value** – Any value of an existing attribute

Returns

unknown

Usage:

```
const items = squared.attr("android.substitute", "viewAttributes" /* string[] */);
items.push("hint", "buttonTint");
/* OR */
squared.attr("android.substitute", "viewAttributes", items.concat(["hint",
↪"buttonTint"]));

squared.attr("android.substitute", "attributeMapping", { "android:src":
↪"app:srcCompat", "icon": "navigationIcon" }); // Set only and does not merge
```

apply(target, options[, setting])

Find extension and merge a configuration object with existing *Extension*.options.

Arguments

- **target** – Name or control of extension
- **options** (object()) – Overriding configuration values
- **setting** (string()) – (optional) Name to use when saving to local storage

Returns

boolean

Usage:

```
squared.apply("android.widget.toolbar", {
  element: {
    "toolbar-id": {
      android: {
        background: "?android:attr/windowBackground"
      },
      appBar: {
        android: {
          theme: "@style/ThemeOverlay.AppCompat.Dark.ActionBar"
        }
      }
    }
  },
  "toolbar-example" // Save
});
```

Alternate:

```
squared.apply("android.widget.toolbar", "toolbar-example"); // Load
```

prefetch(type[, all, ...targets])

Downloads assets to a memory cache which can be used by an *Application* framework. Provides cross-origin support for CSS.

Arguments

- **type** (string()) – *css* | *javascript* | *image* | *svg*
- **all** (boolean()) – (OPTIONAL (targets)) Accept request from any origin
- **targets** – (optional) URL string or root element of a contained Document

Returns

Promise<*PrefetchItem*[]>

Usage:

```
squared.prefetch("css").then(() => squared.parseDocument()); // Cross-origin support

Promise.all(
  squared.prefetch("css", true), // All stylesheets
  squared.prefetch("css", "./undetected.css", document.getElementById("shadow-id").
    ↪ shadowRoot),
  squared.prefetch("svg", "http://example.com/icon.svg", "../images/android.svg")
)
.then(() => squared.parseDocument());
```

parseDocument(...elements)

Starts at the root target element and creates a virtual DOM structure by cascading into all the children. Assets can be preloaded (e.g. *images*) which is required with the *android* framework.

Arguments

- **elements** – (optional) *Element* target by either an id string or *HTMLElement*
- **elements** – (optional) See *ElementSettings*

Returns

- Promise<*Node* | void>
- Promise<*Node*[]> (*multiple*)

Usage:

```
const body = await squared.parseDocument();

const section = await squared.parseDocument({
  element: "section-1",
  projectId: "sqd",
  resourceQualifier: "land",
  enabledIncludes: true
});
```

parseDocumentSync(...elements)

Starts at the root target element and creates a virtual DOM structure by cascading into all the children. No assets are preloaded which is sufficient for the *vdom* framework.

Arguments

- **elements** – (optional) *Element* target by either an id string or *HTMLElement*
- **elements** – (optional) See *ElementSettings*

Returns

- *Node* | undefined
- *Node*[] (*multiple*)

Usage:

```
const body = squared.parseDocumentSync();

const content = squared.parseDocumentSync("content-1");
```

findDocumentNode(target[, all, projectId])

Can be used before saving rendered document to modify auto-generated *Node* attributes.

Arguments

- **target** – *Element* target by either an id string or *HTMLElement* (selectors are supported)
- **all** (boolean()) – (OPTIONAL) Uses filter to return multiple results
- **projectId** (string()) – (optional) Uses an existing project in the current framework

Returns

- *Node* | undefined
- *Node*[] (*all*)

Usage:

```
const body = squared.findDocumentNode(document.body);
const content = squared.findDocumentNode("content-1");

const section1 = squared.findDocumentNode("section"); // Only if no element has id=
↪ "section"
```

(continues on next page)

(continued from previous page)

```
const [section2, section3] = squared.findDocumentNode("main > section", true); // ↳
↳ Always an array with "all"
```

Alternate:

```
const layout = squared.findDocumentNode("relativelayout-1", "project-1"); // ↳
↳ Control id
layout.android("layout_width", "match_parent");
```

latest([*count* = 1])

Get any stored session ids from *parseDocument()* since the last time *clear()* was called.

Arguments

- **count** (number()) – (optional) How many ids at most to be retrieved

Returns

string | string[] | undefined

Usage:

```
squared.parseDocument("id-1", "id-2", "id-3").then(() => {
  // ["00001", "00002", "00003"]
  const id3 = squared.latest();
  const id1 = squared.latest(-1);
  const [id2, id3] = squared.latest(2);
  const [id2, id1] = squared.latest(-2);
});
```

close([*projectId*])

Ends the current session or selected project preventing any further modifications. It is called internally when saving or copying.

Arguments

- **projectId** (string()) – (optional) Targets a project that is not the latest

Returns

Promise<boolean>

Usage:

```
await squared.close(); // Optional
```

reset([*projectId*])

Abandons all stored projects and sets a loaded *Application* to its initial state. The current user settings are retained.

Arguments

- **projectId** (string()) – (optional) Targets a project that is not the latest

Usage:

```
squared.reset(); // Optional
```

save([*projectId*, *timeout*])

Uses the default *Application*.settings to generate an archive of the current session or selected project.

Arguments

- **projectId** (string()) – (optional) Targets a project that is not the default project
- **timeout** (number()) – (optional) Maximum time in seconds for build to complete

ReturnsPromise<[ResponseData](#)>

Usage:

```
squared.save().then(result => console.log(result)); // Default project "_" is used
await squared.save("project-1", 10); // 10 seconds for "project-1" to build
```

Alternate:

```
squared.broadcast(result => { console.log(result.value); }, "111-111-111");
await squared.save("project-1", "111-111-111"); // broadcastId
```

saveAs(filename[, options, setting, overwrite])Save current session or selected project as a new archive using *filename* extension.**Arguments**

- **filename** (string()) – Name of file with a valid archive extension
- **options** (object()) – (optional) See [FileActionOptions](#)
- **setting** (string()) – (optional) Name of setting for local storage
- **overwrite** (boolean()) – (optional) Will not merge previously saved settings with *options*

ReturnsPromise<[ResponseData](#)>

Usage:

```
await squared.parseDocument();
await squared.saveAs("android.zip", { timeout: 15, log: { showSize: true } }); // ⏏
↳ Uses default unnamed project "_"

await squared.parseDocument({ element: document.body, projectId: "project-1" });
await squared.saveAs("project-1.7z", { projectId: "project-1", throwErrors: true });
↳ catch(err => { // Will cancel partial archive download
  console.log(err);
});
```

Alternate:

```
// Save
squared.saveAs("android.zip", { timeout: 15, log: { showSize: true } }, "android-
↳ example"); // Uses own "saveAs" namespace

// Load
squared.saveAs("android.zip", "android-example"); // Any page in same domain
```

appendTo(uri[, options, setting, overwrite])

Save current session or selected project into a copy of an existing archive if found or as a new archive using the file extension.

Arguments

- **uri** (string()) – Location of file with a valid archive extension
- **options** (object()) – (optional) See [FileActionOptions](#)
- **setting** (string()) – (optional) Name of setting for local storage
- **overwrite** (boolean()) – (optional) Will not merge previously saved settings with *options*

ReturnsPromise<[ResponseData](#)>

Usage:

```
squared.parseDocument().then(async () => {
  await squared.appendTo("http://localhost:3000/archives/android.001", { format:
    ↪ "tar" }); // "tar" is explicit (ignored filename)
});

await squared.parseDocument({ element: document.body, projectId: "project-1" });
squared.appendTo("../data/project-1.7z", { projectId: "project-1" }).then(result =>
  ↪ { // Uses NodeJS process.cwd() resolution
    console.log(result);
  });
```

Alternate:

```
// Save
squared.appendTo("./android.zip", { timeout: 20, log: { showSize: true } },
  ↪ "android-example"); // Uses own "appendTo" namespace

// Load
squared.appendTo("./android.zip", "android-example"); // Any page in same domain
```

copyTo(pathname[, options, setting, overwrite])

Save current session or selected project to a local directory.

Arguments

- **pathname** (string()) – Location of a directory accessible to the server process
- **options** (object()) – (optional) See [FileActionOptions](#)
- **setting** (string()) – (optional) Name of setting for local storage
- **overwrite** (boolean()) – (optional) Will not merge previously saved settings with *options*

ReturnsPromise<[ResponseData](#)>

Usage:

```
await squared.parseDocument();
await squared.copyTo("./path/to/project", { timeout: 10, log: { showSize: true } });

squared.parseDocument({ element: document.body, projectId: "project-1" }).then(() =>
  ↪ {
    squared.copyTo("/path/project", { projectId: "project-1" }).then(result => ↪
```

(continues on next page)

(continued from previous page)

```
↪ console.log(result));
});
```

Alternate:

```
// Multiple
squared.copyTo(["/path/project1", "/path/project2"]); // Copies processed files to
↪ "project1" and "project2"

// Save
squared.copyTo("./path/to/project", { timeout: 10, log: { showSize: true } },
↪ "android-example"); // Uses own "copyTo" namespace

// Load
squared.copyTo("./path/to/project", "android-example"); // Any page in same domain
```

saveFiles(filename[, options, setting, overwrite])

Save selected assets as a new archive using *filename* extension.

Arguments

- **filename** (string()) – Name of file with a valid archive extension
- **options** (object()) – (optional) See [FileActionOptions](#)
- **setting** (string()) – (optional) Name of setting for local storage
- **overwrite** (boolean()) – (optional) Will not merge previously saved settings with *options*

Returns

Promise<[ResponseData](#)>

Usage:

```
const options = {
  assets: [
    {
      pathname: "images",
      filename: "android.png",
      uri: "http://localhost:3000/common/images/android.png"
    },
    {
      pathname: "images",
      filename: "android-ldpi.png",
      uri: "http://localhost:3000/common/images/android-ldpi.png"
    },
    {
      pathname: "images",
      filename: "android-hdpi.png",
      uri: "http://localhost:3000/common/images/android-hdpi.png"
    }
  ]
};
await squared.saveFiles("android.zip", options);
```

Alternate:

```
// Save
squared.saveFiles("android.zip", options, "android-example"); // Uses own "saveFiles"
↳ "namespace"

// Load
squared.saveFiles("android.zip", "android-example"); // Any page in same domain
```

appendFiles(uri[, options, setting, overwrite])

Save selected assets into a copy of an existing archive if found or as a new archive using the file extension.

Arguments

- **uri** (string()) – Location of file with a valid archive extension
- **options** (object()) – (optional) See [FileActionOptions](#)
- **setting** (string()) – (optional) Name of setting for local storage
- **overwrite** (boolean()) – (optional) Will not merge previously saved settings with *options*

Returns

Promise<[ResponseData](#)>

Usage:

```
const options = {
  format: "7z",
  assets: [
    {
      pathname: "images",
      filename: "android-xhdpi.png",
      uri: "http://localhost:3000/common/images/android-xhdpi.png"
    },
    {
      pathname: "images",
      filename: "android-xxhdpi.png",
      uri: "http://localhost:3000/common/images/android-xxhdpi.png"
    },
    {
      pathname: "images",
      filename: "android-xxxhdpi.png",
      uri: "http://localhost:3000/common/images/android-xxxhdpi.png"
    }
  ]
};
await squared.appendFiles("http://localhost:3000/archives/android.001", options);
```

Alternate:

```
// Save
squared.appendFiles("android.zip", options, "android-example"); // Uses own
↳ "appendFiles" namespace

// Load
squared.appendFiles("android.zip", "android-example"); // Any page in same domain
```

copyFiles(*pathname*[, *options*, *setting*, *overwrite*])

Save selected assets to a local directory.

Arguments

- **pathname** (string()) – Location of a directory accessible to the server process
- **options** (object()) – (optional) See [FileActionOptions](#)
- **setting** (string()) – (optional) Name of setting for local storage
- **overwrite** (boolean()) – (optional) Will not merge previously saved settings with *options*

Returns

Promise<[ResponseData](#)>

Usage:

```
const options = {
  assets: [
    {
      pathname: "images",
      filename: "android-xhdpi.png",
      uri: "http://localhost:3000/common/images/android-xhdpi.png"
    },
    {
      pathname: "images",
      filename: "android-xxhdpi.png",
      uri: "http://localhost:3000/common/images/android-xxhdpi.png"
    },
    {
      pathname: "images",
      filename: "android-xxxhdpi.png",
      uri: "http://localhost:3000/common/images/android-xxxhdpi.png"
    }
  ]
};
await squared.copyFiles("/path/project", options);
```

Alternate:

```
// Multiple
squared.copyFiles(["/path/project1", "/path/project2"], options); // Copies assets_
↳ to "project1" and "project2"

// Save
squared.copyFiles("./path/to/project", options, "android-example"); // Uses own
↳ "copyFiles" namespace

// Load
squared.copyFiles("./path/to/project", "android-example"); // Any page in same_
↳ domain
```

getElementById(*value*[, *sync*, *cache* = true])

Same behavior as native `document.getElementById`¹ except returns a [Node](#) instance.

¹ <https://developer.mozilla.org/docs/Web/API/Document/getElementById>

Arguments

- **value** (string()) – Case-sensitive match against *Element.id* property
- **sync** (boolean()) – (optional) Will block and not wrap query in a Promise
- **cache** (boolean()) – (optional) Use any existing *Node* instance of *Element*

Returns

- Promise<*Node* | null>
- *Node* | null (*sync*)

Usage:

```
const { element, attributes } = await squared.getElementById("content-id");

const content = squared.getElementById("content-id", true); // Synchronous
const child = content?.find(item => item.elementId === "child-id", { cascade: true }
  =>);
```

querySelector(*selector*[, *sync*, *cache* = true])

Same behavior as native `document.querySelector`² except returns a *Node* instance.

Arguments

- **selector** (string()) – Selector or selectors matching one or more elements
- **sync** (boolean()) – (optional) Will block and not wrap query in a Promise
- **cache** (boolean()) – (optional) Use any existing *Node* instance of *Element*

Returns

- Promise<*Node* | null>
- *Node* | null (*sync*)

Usage:

```
const { element, attributes } = await squared.querySelector("section img[src=vdom.
  <png>");

squared.querySelector("body > p", true)?.each(item => { // Synchronous
  if (item.inline) {
    item.css("display", "inline-block");
  }
});
```

querySelectorAll(*selector*[, *sync*, *cache* = true])

Same behavior as native `document.querySelectorAll`³ except returns an array of *Node* instances.

Arguments

- **selector** (string()) – Selector or selectors matching one or more elements
- **sync** (boolean()) – (optional) Will block and not wrap query in a Promise
- **cache** (boolean()) – (optional) Use any existing *Node* instance of *Element*

Returns

² <https://developer.mozilla.org/docs/Web/API/Document/querySelector>

³ <https://developer.mozilla.org/docs/Web/API/Document/querySelectorAll>

- `Promise<Node[] | null>`
- `Node[] | null (sync)`

Usage:

```
const [img1, img2] = await squared.querySelectorAll("section img");

const children = await squared.getElementById("content-id").querySelectorAll("*");
```

fromElement(*element* [, *sync*, *cache*])

Same behavior as `getElementById()` except it also accepts a native *Element*. The **cache** parameter by default is not enabled.

Arguments

- **element** – *Element* target by either an id string or *HTMLElement*
- **sync** (boolean()) – (optional) Will block and not wrap query in a Promise
- **cache** (boolean()) – (optional) Use any existing *Node* instance of *Element*

Returns

- `Promise<Node[] | null>`
- `Node[] | null (sync)`

Usage:

```
const { element, attributes } = await squared.fromElement(document.body);

const content = squared.fromElement("content-id", true); // Synchronous
const child = content?.find(item => item.elementId === "child-id", { cascade: true }
  ↳);
```

fromNode(*node* [, *sync*, *cache*])

Uses an existing *Node* instance and creates a new instance with any modifications. The **cache** parameter by default is not enabled.

Arguments

- **node** (object()) – See *Node*
- **sync** (boolean()) – (optional) Will block and not wrap query in a Promise
- **cache** (boolean()) – (optional) Use any existing *Node* instance

Returns

- `Promise<Node[] | null>`
- `Node[] | null (sync)`

Usage:

```
let body = await squared.fromElement(document.body);
document.querySelectorAll("body div").forEach(item => document.body.
  ↳removeChild(item));
body = await squared.fromNode(body);

const img = squared.fromNode(body, true).find(item => item.imageElement); //
  ↳Synchronous
```

observe([*enable* = true])

Uses *MutationObserver* to watch for any changes to the *parseDocument()* root element. Start after DOM and third-party libraries are initialized.

Arguments

- **enable** (boolean()) – (optional) Start or stop all root elements who are monitoring

Usage:

```
await squared.parseDocument({
  element: document.body,
  projectId: "project-1",
  observe(mutations, observer, settings) {
    squared.reset("project-1");
    squared.parseDocument(settings).then(() => {
      squared.copyTo("/path/project", { projectId: "project-1", modified: true }).
      →then(result => console.log(result));
    });
  }
});

await squared.copyTo("/path/project", { projectId: "project-1",
  →useOriginalHtmlPage: false }).then(() => {
  squared.observe();
});

squared.observe(false); // Discontinue monitoring for changes
```

Alternate:

```
squared.observe({
  subtree: true,
  childList: true,
  attributes: true,
  characterData: false,
  attributeOldValue: false,
  characterDataOldValue: false
});
```

observeSrc(*targets*[, *callback*, *options*])

Can be used to watch external elements which contain modifiable source files.

Arguments

- **targets** – Elements by either selector or *HTMLElement* (**src** or **href** attribute is required)
- **callback** (function()) – (optional (*options*)) Method to call when a modified event is received
- **options** (object()) – (optional) See *FileObserveOptions*

Returns

Promise<*ObserveSocket* | *ObserveSocket*[]>

Usage:

```
const settings = { element: document.body, projectId: "project-1" };
squared.parseDocument(settings).then(() => {
  squared.observeSrc(
    "link[rel=stylesheet]",
    (ev, element) => {
      squared.reset("project-1");
      squared.parseDocument(settings).then(() => squared.copyTo("/path/project"));
    },
    { port: 8080, secure: false, action: "reload", expires: "1h" } // Optional
  );
});
```

Alternate:

```
await squared.observeSrc("link[rel=stylesheet]"); // Will call location.reload()
```

5.3 android

`android.setViewModel(data[, sessionId = "0"])`

Imports to be used for layout bindings associated by session.

Arguments

- **data** (object()) – See [AppViewModel](#)
- **sessionId** (string()) – (optional) Names a session to be used for storage

Usage:

```
squared.parseDocument(() => {
  android.setViewModel(
    {
      import: ["java.util.Map"],
      variable: [{ name: "map", type: "Map<String, String>" }]
    },
    squared.latest()
  );
});
```

`android.setViewModelByProject(data[, projectId = "_"])`

Imports to be used for layout bindings associated by project.

Arguments

- **data** (object()) – See [AppViewModel](#)
- **projectId** (string()) – (optional) Names a project to be used for storage

Usage:

```
android.setViewModelByProject(
  {
    import: ["java.util.Map"],
    variable: [{ name: "map", type: "Map<String, String>" }]
  }
```

(continues on next page)

(continued from previous page)

```

    },
    "project-1"
  );
  await squared.parseDocument({ element: document.body, projectId: "project-1" });

```

android.addDependency(*group*, *name*[, *version*, *type*])

Include a dependency by group and name to an existing or newly created build.gradle. Uses the default project “_” for storage. Any existing dependency with the same group and name will be overwritten.

Arguments

- **group** (string()) – Namespace of library
- **name** (string()) – Component name in library
- **version** (string()) – (OPTIONAL) Exact version requested
- **type** (number()) – (optional) Dependency namespace method in Gradle

Returns

string

Requirements

- **createBuildDependencies** = *true*

Usage:

```

android.addDependency("androidx.core", "core", "1.12.0"); // Default is
↳ "implementation"
squared.copyTo("/path/project", { dependencyScopes: true }); // Will also include
↳ first-level sub-dependencies (optional)

```

Alternate:

```

android.addDependency("androidx.core", "core", DEPENDENCY_TYPE.COMPILE_ONLY, true);
↳ // Uses latest Maven published release
squared.copyTo("/path/project", { dependencyScopes: "snapshot" });

```

android.addDependencyByProject(*projectId*, *group*, *name*[, *version*, *type*])

Include a dependency to a project by group and name of an existing or newly created build.gradle. Any existing dependency in the project with the same group and name will be overwritten.

Arguments

- **projectId** (string()) – Names a project to be used for storage
- **group** (string()) – Namespace of library
- **name** (string()) – Component name in library
- **version** (string()) – (OPTIONAL) Exact version requested
- **type** (number()) – (optional) Dependency namespace method in Gradle

Returns

string

Requirements

- **createBuildDependencies** = *true*

Usage:

```
android.addDependencyByProject("project-1", "androidx.core", "core", "1.12.0"); //
↳ Default is "implementation"
squared.copyTo("/path/project", { projectId: "project-1", dependencyScopes: true });
↳ // Will also include first-level sub-dependencies (optional)
```

Alternate:

```
android.addDependencyByProject("project-1", "androidx.core", "core", DEPENDENCY_
↳ TYPE.COMPILE_ONLY, true); // Uses latest Maven published release
squared.copyTo("/path/project", { projectId: "project-1", dependencyScopes:
↳ "snapshot" });
```

android.addFontProvider(authority, package, certs, webFonts)

Add additional [Web fonts](#) that can be searched for when resolving first available font family. [Google Fonts](#) is already included.

Arguments

- **authority** (string()) – Class of font provider library
- **package** (string()) – Namespace of font provider
- **certs** (array()) – List of certificates the font provider is signed with
- **webFonts** (string()) – Web font service URL

Returns

boolean | Promise<boolean>

Requirements

- **targetAPI** >= 26
- **createDownloadableFonts** = true

Usage:

```
await android.addFontProvider(
  "com.google.android.gms.fonts",
  "com.google.android.gms",
  ["MIIEqDCCA5CgAwIBAgIJANWFuGx9007...", "MIIEQzCCAyugAwIBAgIJAMLgh0Zk..."],
  "https://www.googleapis.com/webfonts/v1/webfonts?key=1234567890" // Pre-built
↳ JSON object is synchronous
);
```

android.addXmlNs(name, uri)

Aliases of global namespaces for third-party controls used when resolving layout attributes.

Arguments

- **name** (string()) – Prefix to be used with attribute
- **uri** (string()) – Full URL namespace of schema

Usage:

```
android.addXmlNs("tools", "http://schemas.android.com/tools"); // https://developer.
↳ android.com/studio/write/tool-attributes
```

(continues on next page)

(continued from previous page)

```

android.customize(16 /* Jelly Bean */, "ImageView", {
    tools: {
        ignore: "ContentDescription",
        targetApi: "16"
    }
});

```

Output:

```

<FrameLayout xmlns:tools="http://schemas.android.com/tools">
    <ImageView tools:ignore="ContentDescription" tools:targetApi="16">
</FrameLayout>

```

android.customize(*api*, *widget*, *options*)

Global attributes to be applied to every qualifying layout control possibly overwriting any auto-generated attributes.

Arguments

- **api** (number()) – Android SDK build API version
- **widget** (string()) – Namespace of layout control
- **options** (object()) – Attributes to be applied to control

Returns

Record<string, StringMap> | undefined

Requirements

- **customizationsBaseAPI** >= 0
- **customizationsOverwritePrivilege** = true

Usage:

```

android.customize(BUILD_VERSION.ALL /* 0 */, "Button", {
    android: {
        minWidth: "35px",
        minHeight: "25px"
    },
    "_": { // Non-namespaced attributes
        style: "@style/Widget.Material3.Button.TextButton"
    }
});

```

Output:

```

<Button
    android:minWidth="35dp"
    android:minHeight="25dp"
    style="@style/Widget.Material3.Button.TextButton" />

```

android.loadCustomizations(*name*)

Will merge any saved customizations from the same origin. Any previous calls to [android.customize\(\)](#) may be overwritten.

Arguments

- **name** (string()) – Unique identifier to be used for local storage

Usage:

```
squared.settings.targetAPI = BUILD_VERSION.T;
squared.settings.customizationsBaseAPI = 0; // Apply all customizations

android.loadCustomizations("customize-example"); // Any page in same domain

android.customize(BUILD_VERSION.T, "Button", { android: { minWidth: "25px" } });
android.customize(BUILD_VERSION.LATEST, "Button", { android: { minWidth: "30px" } }
→);
```

Output:

```
<Button android:minWidth="25dp" android:minHeight="25dp" />
```

Alternate:

```
squared.settings.targetAPI = BUILD_VERSION.T;
squared.settings.customizationsBaseAPI = [BUILD_VERSION.T, BUILD_VERSION.LATEST];
```

Output:

```
<Button android:minWidth="30dp" android:minHeight="25dp" />
```

android.saveCustomizations(name)

Any valid customizations created using [android.customize\(\)](#) will be saved to local storage.

Arguments

- **name** (string()) – Unique identifier to be used for local storage

Usage:

```
android.customize(BUILD_VERSION.ALL /* 0 */, "Button", {
  android: {
    minWidth: "35px",
    minHeight: "25px"
  }
});
android.customize(BUILD_VERSION.NEXT /* 35 */, "Button", { // Invalid
  android: {
    minWidth: "35px",
    minHeight: "25px"
  }
});
```

android.resetCustomizations()

All customizations currently being used are deleted. Saved customizations in local storage are not affected.

Usage:

```
android.resetCustomizations();
```

android.setResolutionByDeviceName(value)

Sets the resolution when converting browser dimensions into Android device dimensions.

- | | | | |
|------------------|---------------|------------------|--------------------------------|
| • Phone | • Pixel 2 XL | • Pixel 7 | • Nexus 7 |
| • Medium Phone | • Pixel 3 | • Pixel 7a | • Nexus 9 |
| • Foldable | • Pixel 3a | • Pixel 7 Pro | • Nexus 10 |
| • Tablet | • Pixel 3 XL | • Pixel 8 | • TV 4K |
| • Medium Tablet | • Pixel 3a XL | • Pixel 8 Pro | • TV 1080p |
| • Small Desktop | • Pixel 4 | • Pixel C | • Television (4K) |
| • Medium Desktop | • Pixel 4 XL | • Pixel Fold | • Television (1080p) |
| • Desktop | • Pixel 4a | • Nexus 5X | • TV 720p |
| • Large Desktop | • Pixel 5 | • Nexus 6 | • Television (720p) |
| • Pixel | • Pixel 6 | • Nexus 6P | • Automotive |
| • Pixel XL | • Pixel 6a | • Nexus 7 2012 | • Automotive (1024p landscape) |
| • Pixel 2 | • Pixel 6 Pro | • Nexus 7 (2012) | |

The exact configuration for each device can be found in the latest Android Studio.

Arguments

- **name** (string()) – Predefined device name or generic layout

Returns

boolean

Usage:

```
android.setResolutionByDeviceName("Pixel 3a XL");
```

Output:

```
squared.settings.resolutionDPI = 411;  
squared.settings.resolutionScreenWidth = 846;  
squared.settings.resolutionScreenHeight = 560;
```

android.getLocalSettings()

Controller settings which are based on browser defaults can be modified. These are global changes and affect every call to *parseDocument()*.

Returns

ControllerSettingsUI

Usage:

```
const { layout, directory, filename, style, mimeType, unsupported, deviations, ↵  
↵values } = android.getLocalSettings();  
  
layout.fileExtension = ".xml";  
directory.string = "res/values";  
style.buttonBorderStyle = "inset";
```

android.removeObserver(element)

Forwards the target element to *Application* which will stop it from being monitored for changes.

Arguments

- **element** – *HTMLElement* instance

Returns

boolean

Usage:

```
await squared.parseDocument({
  element: document.body,
  observe(mutations, observer, settings) {}
});

android.removeObserver(document.body);
```

5.4 chrome

`chrome.removeObserver(element)`

Forwards the target element to *Application* which will stop it from being monitored for changes.

Arguments

- **element** – *HTMLElement* instance

Returns

boolean

Usage:

```
await squared.parseDocument(document.body);

await squared.copyTo("/path/project", { useOriginalHtmlPage: false, observe: true }
↪).then(() => {
  squared.observe();
});

chrome.removeObserver(document.body);
```


6.1 Android

6.1.1 Interface

```
class NodeUI extends Node {
    static linearData(list: NodeUI[], cleared?: Map<NodeUI, string> | null, absolute?: ↳
↳boolean): LinearData;
    static baseline(list: NodeUI[], text?: boolean, image?: boolean): NodeUI | null;
    static partitionRows(list: NodeUI[], cleared?: Map<NodeUI, string> | null): ↳
↳NodeUI[][];
    alignmentType: number;
    contentAltered: boolean;
    baselineActive: boolean;
    baselineAltered: boolean;
    rendered: boolean;
    excluded: boolean;
    rootElement: boolean;
    lineBreakLeading: boolean;
    lineBreakTrailing: boolean;
    floatContainer: boolean;
    absoluteContainer: boolean;
    renderChildren: NodeUI[];
    renderExtension: Extension<NodeUI>[] | null;
    renderTemplates: NodeTemplate<NodeUI>[] | null;
    documentChildren?: NodeUI[];
    setControlType(controlName: string, containerType?: number): void;
    setExclusions(): void;
    setLayout(): void;
    setAlignment(): void;
    setBoxSpacing(): void;
    attr(name: string, attr: string, value?: string, overwrite?: boolean): string;
    attrx(attr: string, value?: string, overwrite?: boolean): string;
    alignParent(...position: AnchorPositionAttr[]): boolean;
    alignSibling(position: AnchorPositionAttr, documentId?: string): string;
    anchorChain(...values: PositionAttr[]): NodeUI[];
    localizeString(value: string): string;
    inherit(node: NodeUI, ...modules: string[]): PlainObject | null;
    inheritApply(module: string, data: PlainObject): void;
    transferData(node: NodeUI, overwrite?: boolean, ...keys: (string | symbol)[]): void;
```

(continues on next page)

(continued from previous page)

```

clone(id: number): void;
cloneBase(node: NodeUI): void;
is(containerType: number): boolean;
of(containerType: number, ...alignmentType: number[]): boolean;
namespace(name: string): StringMap;
namespaces(): [string, StringMap][];
unsafe(name: string | PlainObject, value?: unknown): unknown;
delete(name: string, ...attrs: string[]): void;
deleteOne(name: string, attr: string): void;
apply(options: PlainObject): void;
lockAttr(name: string, attr: string): void;
unlockAttr(name: string, attr: string): void;
lockedAttr(name: string, attr: string): boolean;
unsetCache(...attrs: (CssStyleAttr | keyof CacheValueUI][]): void;
addAlign(value: number): void;
removeAlign(value: number): void;
hasAlign(value: number): boolean;
hasProcedure(value: number): boolean;
hasResource(value: number): boolean;
hasSection(value: number): boolean;
exclude(options: ExcludeOptions): void;
hide(options?: HideOptions<NodeUI>): NodeTemplate<NodeUI> | null;
replaceTry(options: ReplaceTryOptions<NodeUI>): boolean;
removeTry(options?: RemoveTryOptions<NodeUI>): NodeTemplate<NodeUI> | null;
render(parent: NodeUI): void;
renderEach(predicate: IteratorPredicate<NodeUI, void>): this;
parseWidth(value: string, parent?: boolean): number;
parseHeight(value: string, parent?: boolean): number;
actualRect(position: PositionAttr, dimension?: BoxType): number;
actualPadding(attr: BoxPaddingAttr): number;
actualMargin(attr: BoxMarginAttr): number;
actualBoxWidth(value?: number): number;
actualTextHeight(options?: TextHeightOptions): number;
alignedVertically(siblings?: NodeUI[] | AlignedVerticallyOptions<NodeUI> | null,
cleared?: Map<NodeUI, string> | null, horizontal?: boolean): number;
previousSiblings(options?: TraverseSiblingsOptions): NodeUI[];
nextSiblings(options?: TraverseSiblingsOptions): NodeUI[];
actualSpacing(region: BOX_STANDARD): number;
modifyBox(region: BOX_STANDARD, value: number, negative?: boolean): void;
setBox(region: BOX_STANDARD, options: BoxOptions): void;
getBox(region: BOX_STANDARD): [number, number];
resetBox(region: BOX_STANDARD, node?: NodeUI): void;
transferBox(region: BOX_STANDARD, node: NodeUI): void;
registerBox(region: BOX_STANDARD, node?: NodeUI): NodeUI | null;
setCacheValue(attr: keyof CacheValueUI, value: any): void;
setCacheState(attr: keyof CacheStateUI<NodeUI>, value: any): void;
setCacheStyle(attr: keyof CssStyleMap, value: string): void;
unsetStyle(...attrs: CssStyleAttr[]): void;
extractAttributes(depth?: number, replaceWith?: AnyObject): string;
css(attr: CssStyleAttr, value?: string | null, cache?: boolean): string;
cssSet(attr: CssStyleAttr, value: string, cache?: boolean): string;
translateX(value: number, options?: TranslateOptions): boolean;

```

(continues on next page)

(continued from previous page)

```

    translateY(value: number, options?: TranslateOptions): boolean;
    flex(attr: "inline" | "row" | "column" | "reverse" | "wrap" | "wrapReverse", parent?
↳: boolean | NodeUI, wrapped?: boolean): boolean;
    flex(attr: "alignContent" | "justifyContent" | "basis" | "alignSelf" | "justifySelf",
↳ parent?: boolean | NodeUI, wrapped?: boolean): string;
    flex(attr: "grow" | "shrink" | "order", parent?: boolean | NodeUI, wrapped?:
↳ boolean): number;
    flex<(attr: string, parent?: boolean | NodeUI, wrapped?: boolean): boolean | number
↳ | string;
    getBoxSpacing(region?: BOX_STANDARD): [number, number, number, number];
    getAnchoredSiblings(orientation: OrientationAttr): NodeUI[];
    getPseudoElement(name: PseudoElt | PseudoStyleElt, attr?: CssStyleAttr): CssStyleMap
↳ | string | null;
    isResizable(attr: DimensionSizableAttr, not?: string | string[]): boolean;
    fitToScreen(value: Dimension): Dimension;
    getComments(type: ReferenceType, attrs?: CssStyleAttr[]): [string, string];
    cssValue(attr: CssStyleAttr): string;
    cssValues(...attrs: CssStyleAttr[]): string[];
    set documentRoot(value);
    get documentRoot(): boolean;
    set depth(value);
    get depth(): number;
    set parent(value);
    get parent(): NodeUI | null;
    set documentParent(value);
    get documentParent(): NodeUI;
    set containerName(value);
    get containerName(): string;
    set autoPosition(value);
    get autoPosition(): boolean;
    set inlineText(value);
    get inlineText(): boolean;
    set textContent(value);
    get textContent(): string;
    get styleText(): boolean;
    set multiline(value);
    get multiline(): boolean;
    set visible(value);
    get visible(): boolean;
    set controlName(value);
    get controlName(): string;
    set actualParent(value);
    get actualParent(): NodeUI | null;
    set siblingsLeading(value);
    get siblingsLeading(): NodeUI[];
    set siblingsTrailing(value);
    get siblingsTrailing(): NodeUI[];
    set renderParent(value);
    get renderParent(): NodeUI | null;
    set outerWrapper(value);
    get outerWrapper(): NodeUI | null;
    set companion(value);

```

(continues on next page)

(continued from previous page)

```

get companion(): NodeUI | null;
set renderedAs(value);
get renderedAs(): NodeTemplate<NodeUI> | null;
set horizontalRows(value);
get horizontalRows(): NodeUI[][] | null;
set containerType(value: number);
get containerType(): number;
set positioned(value);
get positioned(): boolean;
set controlId(name: string);
get controlId(): string;
get referenceId(): string;
get documentId(): string;
set textIndent(value);
get textIndent(): number;
get preIndent(): [string, NodeUI] | null;
get causesLineBreak(): boolean;
set renderExclude(value: boolean);
get renderExclude(): boolean;
set renderAs(value);
get renderAs(): NodeUI | null;
set labelFor(value);
get labelFor(): NodeUI | null;
set innerWrapped(value);
get innerWrapped(): NodeUI | null;
set use(value);
get use(): string;
set localSettings(value);
get localSettings(): LocalSettingsUI;
get extensions(): string[];
get scrollElement(): boolean;
get controlElement(): boolean;
get imageContainer(): boolean;
get baselineHeight(): number;
get support(): SupportUI;
get layoutElement(): boolean;
get layoutHorizontal(): boolean;
get layoutVertical(): boolean;
get nodeGroup(): boolean;
get tagDisplay(): string;
get inlineVertical(): boolean;
get inlineDimension(): boolean;
get blockStatic(): boolean;
get blockVertical(): boolean;
get blockDimension(): boolean;
get inlineFlow(): boolean;
get verticalAligned(): boolean;
get variableWidth(): boolean;
get variableHeight(): boolean;
get fullWidth(): boolean;
get fullHeight(): boolean;
get fullContentWidth(): boolean;

```

(continues on next page)

(continued from previous page)

```

    get fullContentHeight(): boolean;
    get positiveAxis(): boolean;
    get leftTopAxis(): boolean;
    get baselineElement(): boolean;
    get flowElement(): boolean;
    get flowContent(): boolean;
    get flowChildren(): NodeUI[];
    get flexRow(): boolean;
    get flexColumn(): boolean;
    get previousSibling(): NodeUI | null;
    get nextSibling(): NodeUI | null;
    get firstStaticChild(): NodeUI | null;
    get lastStaticChild(): NodeUI | null;
    get onlyChild(): boolean;
    get onlyStaticChild(): boolean;
    get horizontalRowStart(): boolean;
    get horizontalRowEnd(): boolean;
    get innerBefore(): NodeUI | null;
    get innerAfter(): NodeUI | null;
    get rendering(): boolean;
    get boxReset(): number[];
    get boxAdjustment(): number[];
    get overflowX(): boolean;
    get overflowY(): boolean;
    get textEmpty(): boolean;
    get textWidth(): number;
    get textMetrics(): TextMetrics | null;
    get innerMostWrapped(): NodeUI;
    get outerMostWrapper(): NodeUI;
    get firstLineStyle(): CssStyleMap | null;
    get firstLetterStyle(): CssStyleMap | null;
    get textAlignLast(): string;
    get textJustified(): boolean;
    get alignContent(): "start" | "end" | "center" | "baseline" | "";
    get transformValue(): string;
    get outerRegion(): BoxRectDimension;
}

class View extends NodeUI {
    static availablePercent(nodes: View[], dimension: DimensionAttr, boxSize: number, ↵
    ↵ flexContainer?: boolean): number;
    static getControlName(containerType: number, api?: number): string;
    api: number;
    materialDesign: boolean;
    android(attr: string, value?: string, overwrite?: boolean): string;
    app(attr: string, value?: string, overwrite?: boolean): string;
    clone(id?: number, options?: CloneOptions): View;
    applyOptimizations(): boolean;
    applyCustomizations(overwrite?: boolean): void;
    formatted(value: string, overwrite?: boolean): void;
    mergeGravity(attr: LayoutGravityAttr, alignment: LayoutGravityDirectionAttr, ↵
    ↵ overwrite?: boolean): void;

```

(continues on next page)

(continued from previous page)

```

    anchor(position: AnchorPositionAttr, documentId?: string, overwrite?: boolean): ↵
↵boolean;
    anchorChain(...values: PositionAttr[]): View[];
    anchorParent(orientation: OrientationAttr, bias: number, overwrite: true): boolean;
    anchorParent(orientation: OrientationAttr, bias?: number, style?: LayoutChainStyle, ↵
↵overwrite?: boolean): boolean;
    anchorStyle(orientation: OrientationAttr, bias: number, overwrite: true): void;
    anchorStyle(orientation: OrientationAttr, bias: number, style?: LayoutChainStyle, ↵
↵overwrite?: boolean): void;
    anchorDelete(...position: AnchorPositionAttr[]): void;
    anchorClear(update?: View | boolean, renderParent?: View | null): void;
    deprecated(attr: string, value: string, output: PlainObject): boolean | undefined;
    supported(attr: string, value: string, output: PlainObject): boolean;
    combine(sort: boolean, ...values: string[]): string[];
    combine(...values: string[]): string[];
    setLayoutWidth(value: string, overwrite?: boolean): void;
    setLayoutHeight(value: string, overwrite?: boolean): void;
    setLayoutPercent(value: string, horizontal?: boolean): void;
    setSingleLine(maxLines: boolean, ellipsis?: boolean): void;
    setConstraintDimension(percentAvailable?: number): number;
    setFlexDimension(dimension: DimensionAttr, percentAvailable?: number, weight?: ↵
↵number): number;
    getMatchConstraint(parent?: View, container?: boolean, percent?: boolean): string;
    getAnchorPosition(parent: View, horizontal: boolean, modifyAnchor?: boolean): Partial
↵<BoxRect>;
    isUnstyled(checkMargin?: boolean): boolean;
    isAnchored(options: IsAnchoredOptions): boolean;
    getHorizontalBias(rect?: BoxRect): number;
    getVerticalBias(rect?: BoxRect): number;
    getAbsolutePaddingOffset(region: number, value: number): number;
    getMarginPercent(horizontal: boolean): number;
    getPaddingPercent(horizontal: boolean): number;
    hasFlex(direction: LayoutDirectionAttr): boolean | 0;
    hasFixedDimension(dimension: DimensionAttr): boolean;
    set anchored(value);
    get anchored(): boolean;
    set localSettings(value);
    get localSettings(): LocalSettingsUI;
    set useSystemColors(value);
    get useSystemColors(): boolean;
    get documentId(): string;
    get anchorTarget(): View;
    get constraint(): Constraint<View>;
    get layoutFrame(): boolean;
    get layoutLinear(): boolean;
    get layoutGrid(): boolean;
    get layoutRelative(): boolean;
    get layoutConstraint(): boolean;
    get layoutFragment(): boolean;
    get layoutAnchoring(): boolean;
    get layoutWidth(): string;
    get layoutHeight(): string;

```

(continues on next page)

(continued from previous page)

```

get inlineWidth(): boolean;
get inlineHeight(): boolean;
get blockWidth(): boolean;
get blockHeight(): boolean;
get flexibleWidth(): boolean;
get flexibleHeight(): boolean;
get flexibleHorizontal(): boolean;
get watch(): WatchInterval | undefined;
get tasks(): TaskCommand[] | undefined;
get target(): HTMLElement | null;
get support(): SupportUI;
}

```

Added in version 5.2.1:

- *NodeUI* property getter **textMetrics** for font dimensions was created.

Added in version 5.2.0:

- *NodeUI* property getter **tagDisplay** for rendered behavior was created.
- *NodeUI* property getter **alignContent** for vertical layout position was created.
- *NodeUI* method **extractAttributes** *optional* argument **replaceWith** as *AnyObject* was created.
- *NodeUI* method **getPseudoElement** argument **name** with *PseudoStyleElt* was amended.
- *NodeUI* method **flex** argument **parent** with *NodeUI* was amended.
- *View* property getter **useSystemColors** for device color translation was created.

Deprecated since version 5.2.0:

- *NodeUI* property getter **flexRow** as *NodeUI.flex("row")* is pending removal.
- *NodeUI* property getter **flexColumn** as *NodeUI.flex("column")* is pending removal.

6.1.2 Extensions

Layout rendering can be fully customized as the program was built to be nearly completely modular. Some of the common layouts already have built-in extensions which you can load or unload based on your preference.

Interface

Listing 1: squared.accessibility

```

interface ExtensionAccessibilityOptions {
    displayLabel: boolean;
}

```

Listing 2: squared.grid

```

interface ResourceGridOptions {
    floatPrecision: number;
}

```

Listing 3: squared.css-grid

```
interface ResourceCssGridOptions {  
    floatPrecision: number;  
}
```

Listing 4: squared.list

```
interface ExtensionListOptions {  
    ordinalFontSizeAdjust: number;  
    ordinalPaddingLeft: number;  
    imagePaddingRight: number;  
    /* base */  
    symbolDisc: string;  
    symbolSquare: string;  
    symbolCircle: string;  
    symbolDisclosureOpen: string;  
    symbolDisclosureClosed: string | [string, string /* rtl */];  
    symbolFallback: string;  
    markerStyle: CssStyleMap;  
}
```

Listing 5: android.substitute

```
interface ExtensionSubstituteOptions {  
    element: Record<string, ExtensionAttributeElement>;  
    resource: PlainObject;  
    viewAttributes: string[];  
    attributeMapping: StringMap;  
}
```

Listing 6: android.delegate.multiline

```
interface DelegateMultilineOptions {  
    mergeSingleLine: boolean;  
}
```

Listing 7: android.delegate.scrollbar

```
interface DelegateScrollbarOptions {  
    alwaysDrawTrack: boolean;  
    alwaysDrawHorizontalTrack: boolean;  
    alwaysDrawVerticalTrack: boolean;  
    style: "none" | "outsideInset" | "insideInset" | "insideOverlay" | "outsideOverlay";  
    size: string; // px  
    thinSize: string;  
    fadeDuration: number;  
    delayBeforeFade: number;  
}
```

Listing 8: android.resource.background

```
interface ResourceBackgroundOptions {  
    outlineAsInsetBorder: boolean;  
    enableImageRepeat: boolean;  
}
```

Listing 9: android.resource.dimens

```
interface ResourceDimensOptions {  
    percentAsResource: boolean;  
    floatPrecision: number;  
}
```

Listing 10: android.resource.fonts

```
interface ResourceFontsOptions {  
    defaultFontFamily: string;  
    systemFonts: string[];  
    disableFontAlias: boolean;  
    installGoogleFonts: boolean;  
    fontSizeAdjust: number;  
    floatPrecision: number;  
}
```

Listing 11: android.resource.fragment

```
interface ExtensionFragmentOptions {  
    viewAttributes: string[];  
    viewAttributesApp: string[];  
    retainAttributes: string[];  
    retainAttributesApp: string[];  
    dynamicNestedFragments: boolean;  
}
```

Listing 12: android.resource.includes

```
interface ExtensionIncludesOptions {  
    viewAttributes: string[];  
    viewAttributesApp: string[];  
    viewAttributesOuterView: string[];  
}
```

Listing 13: android.resource.strings

```
interface ResourceStringsOptions {  
    numberAsResource: boolean;  
}
```

Listing 14: android.resource.svg

```
interface ResourceSvgOptions {
    textAsImage: boolean;
    mergeClipPath: boolean;
    mergeVectorDrawable: boolean;
    dependencyVectorDrawable: "animated" | "seekable";
    transformExclude: SvgTransformExclude;
    animateInterpolator: string;
    floatPrecision: number;
    floatPrecisionKeyTime: number;
}
```

Listing 15: jetpack.compose.view

```
interface JetpackComposeViewOptions {
    viewAttributes: string[];
    renderChildren: boolean;
}
```

Note: These are the built-in extensions which have configurable settings.

Example usage

Some extensions have a few settings which can be configured. The default settings usually achieve the best overall rendering accuracy without noticeably affecting performance.

Listing 16: Create

```
class Sample extends squared.base.ExtensionUI {
    options = {
        attributeName: [],
        floatPrecision: 3
    };

    constructor(name, framework = 0, options = {}) {
        super(name, framework, options);
    }

    processNode(node) {
        const data = this.project.get(node.element, node.localSettings.projectId);
        if (data) {
            node.each((child, index) => child.element.title = data[index]);
        }
    }
}
```


Listing 17: Install

```
const sample = new Sample("widget.example.com", 2 /* APP_FRAMEWORK.ANDROID */, {
  ↪tagNames: ["span", "li"], dependencies: ["android.substitute"] });
squared.add(sample);
// OR
squared.add([sample, { attributeName: ["width", "height"] }]);
```

Listing 18: Configure

```
squared.attr("widget.example.com", "floatPrecision", 2); // typeof is enforced and will
↪only set existing attributes
```

Listing 19: Add project data

```
const ext = squared.get("widget.example.com");

ext.project.set(element, await fetch(url?id=1)); // Map interface with optional
↪"projectId" argument
ext.project.set(element, await fetch(url?id=2), "project-1");

const data = ext.project.get(element, "project-2"); // Returns data from default project
↪(id=1)
```

Added in version 5.2.1:

- *ResourceSvgOptions* properties **dependencyVectorDrawable** as *string* was created.

Added in version 5.2.0:

- *ExtensionListOptions* properties were created:

- | | | |
|----------------|--------------------------|---------------|
| – symbolDisc | – symbolDisclosureOpen | – markerStyle |
| – symbolSquare | – symbolDisclosureClosed | |
| – symbolCircle | – symbolFallback | |

- *DelegateScrollbarOptions* properties were created:

- | | | |
|-----------------------------|----------------|-------------------|
| – alwaysDrawHorizontalTrack | – size | – delayBeforeFade |
| – alwaysDrawVerticalTrack | – thinSize | |
| – style | – fadeDuration | |

- *ResourceSvgOptions* properties **mergeClipPath** | **mergeVectorDrawable** as *boolean* were created.
- *ProjectMap* methods **get** | **has** will also check default project “_” for key.

6.1.3 Data Binding

View model data can be applied to most HTML elements using the *dataset*¹ feature. Different view models can be used for every *parseDocument()* session.

Leaving the *sessionId* empty uses the default view model which is searched last for all projects when attempting a bind.

Method

```
squared.parseDocument("element-1", "element-2", "element-3").then(nodes => {
  const sessions = squared.latest(2); // ["2", "3"]
  android.setViewModel(
    {
      import: ["java.util.Map", "java.util.List"],
      variable: [
        { name: "user", type: "com.example.User" },
        { name: "list", type: "List<String>" },
        { name: "map", type: "Map<String, String>" },
        { name: "index", type: "int" },
        { name: "key", type: "String" }
      ]
    },
    sessions[0] // nodes[1].sessionId
  );
  android.setViewModel(
    {
      import: ["java.util.Map"],
      variable: [
        { name: "map", type: "Map<String, String>" }
      ]
    },
    sessions[1] // nodes[2].sessionId
  );
});
```

```
squared.parseDocument({
  element: "main",
  enabledViewModel: true,
  dataBindableElements: [
    {
      selector: "#first_name",
      namespace: "android", // "android" is default
      attr: "text",
      expression: "user.firstName"
    },
    {
      selector: "#last_name",
      attr: "text",
      expression: "user.lastName"
    }
  ]
});
```

(continues on next page)

¹ <https://developer.mozilla.org/docs/Web/API/HTMLElement/dataset>

(continued from previous page)

```

        selector: "#remember_me",
        attr: "checked",
        expression: "user.rememberMe",
        twoWay: true
    }
],
data: {
    viewModel: {
        import: ["java.util.List"],
        variable: [
            { name: "user", type: "com.example.User" }
        ]
    }
}
});

```

Inline

Creating a view model inline can be more convenient for simple layouts. JavaScript is the recommended solution when `parseDocument()` is called multiple times.

```
data-viewmodel-{namespace}-{attribute} -> data-viewmodel-android-text
```

These two additional output parameters are required when using the **data-viewmodel** prefix.

```

<div id="main">
  <label>Name:</label>
  <input id="first_name" type="text" data-viewmodel-android-text="user.firstName" />
  <input id="last_name" type="text" data-viewmodel-android-text="user.lastName" />
  <input id="remember_me" type="checkbox" data-viewmodel-android-checked="=user.
  rememberMe" />
</div>

```

Tip: Use “=” to create a two-way binding.

Listing 20: Output

```

<layout>
  <data>
    <import type="java.util.Map" />
    <import type="java.util.List" />
    <variable name="user" type="com.example.User" />
    <variable name="list" type="List<String>" />
    <variable name="map" type="Map<String, String>" />
    <variable name="index" type="int" />
    <variable name="key" type="String" />
  </data>
  <LinearLayout android:id="@+id/main">
    <TextView android:text="Name:" />
    <EditText

```

(continues on next page)

(continued from previous page)

```

        android:id="@+id/first_name"
        android:inputType="text"
        android:text="@{user.firstName}" />
    <EditText
        android:id="@+id/last_name"
        android:inputType="text"
        android:text="@{user.lastName}" />
    <CheckBox
        android:id="@+id/remember_me"
        android:checked="@={user.rememberMe}" />
</LinearLayout>
</layout>

```

6.1.4 Resources

Projects can include additional values which may have missed detection and are required during compilation.

Interface

```

class Resource {
    static addTheme(resourceId: number, theme: ThemeAttribute, options?: ␣
↳ AddThemeOptions): boolean;
    static addString(resourceId: number, value: string, name?: string, options?: ␣
↳ AddStringOptions): string;
    static addArray(resourceId: number, name: string, items: string[], options?: ␣
↳ AddArrayOptions): string;
    static addImage(resourceId: number, images: StringMap, options?: AddImageOptions): ␣
↳ string;
    static addColor(resourceId: number, value: ColorRGB | string, transparency?: ␣
↳ boolean): string;
    static addDimen(resourceId: number, name: string, value: string): string;
}

```

Example usage

```

squared.parseDocument().then(node => {
    const resourceId = node.localSettings.resourceId;
    android.base.Resource.addString(resourceId, "123", "title", { isText: true });
    android.base.Resource.addArray(resourceId, "list_1", ["1", "2", "3"], { type: 8 }); // ␣
↳ RESOURCE.STRING_ARRAY
    android.base.Resource.addColor(resourceId, "#000000");
    android.base.Resource.addDimen(resourceId, "border_width", "1px");
    android.base.Resource.addTheme(resourceId,
        {
            name: "AppTheme",
            items: { "android:windowTranslucentStatus": "true" }
        },
        { pathname: "res/values", filename: "theme.xml" }
    );
});

```

(continues on next page)

(continued from previous page)

```
);
});
```

6.1.5 Layout Includes

Some applications can benefit from using includes or merge tags to share common templates. Nested includes are also supported.

Method

```
android.setViewModelByProject(
{
    variable: [{ name: "exampleData", type: "com.example.ExampleData" }]
},
"project-1"
);

squared.parseDocument({
    element: document.body,
    projectId: "project-1", // Affects all layouts in same project
    enabledIncludes: true,
    includableElements: [
        {
            selectorStart: "#item2",
            selectorEnd: "#item4",
            pathname: "app/src/main/res/layout-land",
            filename: "filename1.xml",
            merge: true // Multiple elements will auto-merge
        },
        {
            selectorStart: "#item5",
            selectorEnd: "#item5",
            filename: "filename2", // Uses default layout file extension ".xml"
            viewModel: "exampleData" // One element only when "merge = false"
        }
    ]
});
```

Note: By *sessionId* has precedence when associating a view model.

Inline

```
<div>
  <div id="item1">Item 1</div>
  <div id="item2" data-android-include-start="true" data-android-include-merge="true"
↳data-pathname-android="app/src/main/res/layout-land" data-filename-android="filename1.
↳xml">Item 2</div>
  <div id="item3">Item 3</div>
  <div id="item4" data-android-include-end="true">Item 4</div>
  <div id="item5" data-android-include="filename2" data-android-include-end="true" data-
↳android-include-viewmodel="exampleData">Item 5</div>
</div>
```

Tip: data-pathname-android AND data-filename-android can also be used with any *parseDocument()* root element.

Listing 21: Output

```
<!-- res/layout/activity_main.xml -->
<LinearLayout>
  <TextView>Item 1</TextView>
  <include layout="@layout/filename1" />
  <include layout="@layout/filename2" app:exampleData="@{exampleData}" />
</LinearLayout>

<!-- res/layout-land/filename1.xml -->
<merge>
  <TextView>Item 2</TextView>
  <TextView>Item 3</TextView>
  <TextView>Item 4</TextView>
</merge>

<!-- res/layout/filename2.xml -->
<layout>
  <data>
    <variable name="exampleData" type="com.example.ExampleData" />
  </data>
  <TextView>Item 5</TextView>
</layout>
```

The attributes data-android-include-start and data-android-include-end can only be applied to elements which share the same parent container.

See also:

Demo page using *squared-express*¹ for an actual implementation.

¹ <https://github.com/anpham6/squared/blob/master/html/demos/gradient.html>

6.1.6 Substitute

There are some cases where a DOM layout tree has to be transformed into an equivalent layout using third-party controls.

Method

```
await squared.parseDocument({
  element: document.body,
  include: ["android.substitute"],
  substitutableElements: [{
    selector: "#navigation",
    tag: "com.google.android.material.tabs.TabLayout",
    tagChild: "com.google.android.material.tabs.TabItem",
    tagChildAttr: {
      android: {
        layout_height: "match_parent"
      }
    },
    renderChildren: true,
    autoLayout: true
  }]
});
const navigation = squared.findDocumentNode("navigation");
navigation.android("layout_height", "match_parent");
```

Tip: Use `node.lockAttr(namespace, attribute)` when your changes are overridden by an extension.

Inline

```
<ul id="navigation"
  data-use-android="android.substitute"
  data-android-attr="layout_height::match_parent"
  data-android-substitute-tag="com.google.android.material.tabs.TabLayout"
  data-android-substitute-tag-child="com.google.android.material.tabs.TabItem"
  data-android-substitute-tag-child-attr="layout_height::match_parent"
  data-android-substitute-auto-layout="true">
  <li>TAB 1</li>
  <li>TAB 2</li>
  <li>TAB 3</li>
</ul>
```

Listing 22: Output

```
<com.google.android.material.tabs.TabLayout
  android:id="@+id/navigation"
  android:layout_height="match_parent"
  android:layout_width="match_parent">
  <com.google.android.material.tabs.TabItem
```

(continues on next page)

(continued from previous page)

```

        android:layout_height="match_parent"
        android:layout_width="wrap_content"
        android:text="@string/tab_1" />
<com.google.android.material.tabs.TabItem
    android:layout_height="match_parent"
    android:layout_width="wrap_content"
    android:text="@string/tab_2" />
<com.google.android.material.tabs.TabItem
    android:layout_height="match_parent"
    android:layout_width="wrap_content"
    android:text="@string/tab_3" />
</com.google.android.material.tabs.TabLayout>

```

6.1.7 Jetpack Compose

Most mobile applications do not have a deeply nested hierarchy and are generally better to implement using declarative programming.

Method

```

squared.settings.composableElements = ["main", "#content", "--boxShadow", "--height=300px",
↪];
squared.settings.createBuildDependencies = true; // Optional

```

You can also do it using the `android.substitute` extension directly inside the HTML element.

```

squared.add(["android.substitute", {
    element: {
        content: { android: { layout_width: "match_parent" } }
    }
}]);

const items = squared.attr("android.substitute", "viewAttributes");
items.push("hint", "buttonTint");
squared.attr("android.substitute", "attributeMapping", {
    "android:src": "app:srcCompat", // Rename
    "icon": "navigationIcon" // Set "android:icon"
});

squared.parseDocument({
    element: document.body,
    substitutableElements: [{
        selector: "#content",
        tag: "androidx.compose.ui.platform.ComposeView",
        renderChildren: false
    }],
    enabledSubstitute: true, // Disabled built-in extensions have convenience toggles
    /* OR */
    include: ["android.substitute"]
});

```


Inline

```
<body>
  <header style="height: 100px"></header>
  <main id="content"
    data-use="android.substitute"
    data-android-substitute-tag="androidx.compose.ui.platform.ComposeView"
    style="height: 300px; box-shadow: 10px 5px 5px black;">
    <!-- Interior elements are not rendered -->
  </main>
  <footer style="height: 80px"></footer>
</body>
```

Tip: Compose will remove child elements by default. You can preserve them by explicitly using the **renderChildren** property.

```
<div id="fragment"
  data-use="android.substitute"
  data-android-substitute-tag="androidx.fragment.app.FragmentContainerView"
  data-android-substitute-render-children="false"
  data-android-attr="name::com.github.fragment;tag::example">
  <!-- Interior elements are not rendered -->
</div>
```

Tip: You can also use `android.substitute` to create fragments within a layout similar to Compose.

6.1.8 Modify Attributes

System or extension generated attributes can be overridden preceding finalization. They will only be visible on the declared framework.

Tip: Default namespace is `android`.

Settings

```
squared.settings.showAttributes = {
  "hyphenationFrequency": "full", // Replace all
  "android:fontFeatureSettings": null, // Delete all
  "app:menu": [
    "@menu/menu_1", "@menu/menu_2", // Replace with "@menu/menu_2" when value is "@menu/
↪ menu_1"
    "@menu/menu_3", null // Delete attribute when value is "@menu/menu_3"
  ]
};
```

Method

```
squared.parseDocument().then(() => {
  const node = squared.findDocumentNode("customId");
  node.android("layout_width", "match_parent");
  node.android("layout_height", "match_parent");
  node.app("layout_scrollFlags", "scroll|exitUntilCollapsed");
});
```

Inline

```
data-android-attr-{namespace}?
```

```
<div id="customId"
  data-android-attr="layout_width::match_parent;layout_height::match_parent"
  data-android-attr-app="layout_scrollFlags::scroll|exitUntilCollapsed">
</div>
```

Listing 23: Output

```
<LinearLayout
  android:id="@+id/customId"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  app:layout_scrollFlags="scroll|exitUntilCollapsed" />
```

6.1.9 Downloadable Fonts

Google Fonts¹ are pre-installed and can be used without any additional configuration. Here is what happens when these settings are enabled:

```
squared.settings.outputDocumentEditing = true;
squared.settings.createManifest = true;
squared.settings.createBuildDependencies = true;
```

Enable

Listing 24: Install and build^{Page 70, 2}

```
await android.addFontProvider(
  "com.google.android.gms.fonts",
  "com.google.android.gms",
  ["MIIEqDCCA5CgAwIBAgIJANWFuGx9007...", "MIIEQzCCAyugAwIBAgIJAMLgh0Zk..."],
  "https://www.googleapis.com/webfonts/v1/webfonts?key=1234567890" // JSON object is_
  ↪ synchronous
);
```

(continues on next page)

¹ <https://developer.android.com/guide/topics/ui/look-and-feel/downloadable-fonts>

² https://developers.google.com/fonts/docs/developer_api

(continued from previous page)

```
squared.parseDocument({ element: document.body, createDownloadableFonts: true }).then(() ↵
↵=> squared.saveAs("fonts.zip"));
```

Output

Listing 25: AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
  <application android:theme="@style/AppTheme">
    <meta-data android:name="preloaded_fonts" android:resource="@array/preloaded_fonts
↵" />
  </application>
</manifest>
```

Listing 26: build.gradle

```
dependencies {
  implementation 'androidx.appcompat:appcompat:1.6.0'
}
```

Disable

```
squared.attr("android.resource.fonts", "installGoogleFonts", false); // Use browser and ↵
↵local fonts only
```

6.1.10 Multiline

Browser to device measurements are not expected to be accurate due to issues with font availability and screen resolutions. Text wrapping can be adjusted at the global and element level.

Global

```
squared.settings.fontMeasureAdjust = 1; // 1px added per character to width
```

Inline

Listing 27: Source

```
<p>
  content content content content
  content content content content
</p>
```

Listing 28: Wrap late < 0

```
<p data-android-font-measure-adjust="-0.5">
  content content content content content
  content content content
</p>
```

Listing 29: Wrap early > 0

```
<p data-android-font-measure-adjust="1">
  content content content
  content content content
  content content
</p>
```

Listing 30: No adjustment = 0

```
<p data-android-font-measure-adjust="0">
  content content content content
  content content content content
</p>
```

Listing 31: Ignore

```
<p data-android-font-measure-adjust="false">
  content content content content content content content content
</p>
```

Tip: CSS style property `initial-letter` uses the *multiline* extension for detection and rendering.

6.1.11 SVG Animations

Initialize

Animations are usually started before *DOMContentLoaded* has been fired and `parseDocument()` may be unable to read the initial **attribute** values. Inline hints can be used to recreate the non-CSS initial values when they are altered by the native SVG.

Listing 32: JSON

```
<svg viewBox="0 0 200 200">
  <rect id="animate" x="10" y="10" width="30" height="20" fill="purple" data-base-value='
  ↳ { "x": 10, "y": 10, "width": 30, "height": 20, "fill": "purple" }'>
    <animate attributeName="y" values="150; 75; 0; 50; 100;" keyTimes="0; 0.2; 0.5; 0.7;
    ↳ 1" begin="0s" dur="9s" repeatCount="indefinite" />
  </rect>
</svg>
```

Listing 33: Delimited

```
<svg width="120" height="120" viewBox="0 0 240 240" data-base-value="width: 120; height: 120; opacity: 1;">
  <rect x="10" y="0" width="100" height="50" stroke="black" stroke-width="2" fill="purple" data-base-value="x: 10; y: 0; width: 100; fill: purple; opacity: 1; stroke-opacity: 1; fill-opacity: 1;">
    <animate attributeName="x" values="0; 100;" keyTimes="0; 1" begin="0s" dur="10s" repeatCount="indefinite" />
    <animate attributeName="y" values="0; 100" keyTimes="0; 1" begin="0s; 5s; 10s" dur="10s" repeatCount="1" />
  </rect>
</svg>
```

Play

Only the XML based layout and resource files can be viewed on the Android device/emulator without any Java/Kotlin backend code. To play animations in the emulator you also have to *start* the animation in *MainActivity.java*.

```
import android.graphics.drawable.Animatable;

android.widget.ImageView svg = findViewById(R.id.imageview_1);
if (svg != null) {
    Animatable animatable = (Animatable)svg.getDrawable();
    animatable.start();
}
```

6.1.12 Exclusions

Most attributes can be excluded from the generated XML. One or more exclusions can be applied to any element using the OR “|” operator.

Tip: Defining an element `id` will prevent it from being removed during the optimization phase.

Interface

Listing 34: squared.lib.base.constant

```
enum APP_SECTION {
    DOM_TRAVERSE = 1,
    EXTENSION = 2,
    RENDER = 4,
    ALL = DOM_TRAVERSE | EXTENSION | RENDER
}

enum NODE_RESOURCE {
    BOX_STYLE = 1,
    BOX_SPACING = 2,
```

(continues on next page)

(continued from previous page)

```

    FONT_STYLE = 4,
    VALUE_STRING = 8,
    IMAGE_SOURCE = 16,
    ASSET = FONT_STYLE | VALUE_STRING | IMAGE_SOURCE,
    ALL = BOX_STYLE | BOX_SPACING | ASSET
}

enum NODE_PROCEDURE {
    CONSTRAINT = 1,
    LAYOUT = 2,
    ALIGNMENT = 4,
    ACCESSIBILITY = 8,
    LOCALIZATION = 16,
    CUSTOMIZATION = 32,
    OPTIMIZATION = 64,
    DOCUMENT = CONSTRAINT | LAYOUT | ALIGNMENT,
    ENHANCEMENT = ACCESSIBILITY | CUSTOMIZATION | OPTIMIZATION,
    USABILITY = ENHANCEMENT | LOCALIZATION,
    ALL = DOCUMENT | USABILITY
}

```

Method

```

const { NODE_RESOURCE, NODE_PROCEDURE, APP_SECTION } = squared.base.lib.constant;

await squared.parseDocument({
  element: document.body,
  excludeQuery: [
    {
      selector: "h1",
      section: APP_SECTION.DOM_TRAVERSE | APP_SECTION.EXTENSION,
      resource: NODE_RESOURCE.BOX_SPACING | NODE_RESOURCE.FONT_STYLE,
      procedure: NODE_PROCEDURE.CONSTRAINT | NODE_PROCEDURE.LOCALIZATION | NODE_
↪PROCEDURE.CUSTOMIZATION | NODE_PROCEDURE.OPTIMIZATION
    },
    {
      selector: "p > span",
      resource: NODE_RESOURCE.FONT_STYLE
    },
    {
      selector: "#cb1",
      procedure: NODE_PROCEDURE.ACCESSIBILITY
    }
  ]
});

```

Inline

```
<h1>
  data-exclude-section="DOM_TRAVERSE | EXTENSION"
  data-exclude-resource="BOX_SPACING | FONT_STYLE"
  data-exclude-procedure="CONSTRAINT | LOCALIZATION | CUSTOMIZATION | OPTIMIZATION">
    title
</h1>
<p>
  <span data-exclude-resource="FONT_STYLE">content</span>
  <input id="cb1" type="checkbox" data-exclude-procedure="ACCESSIBILITY"><label for="cb1
  ↪">checkbox text</label>
</p>
```

Warning: These may cause warnings when you compile your project and should only be used when an extension has their custom attributes overwritten.

6.1.13 Miscellaneous

System Colors

Any `R.color` that starts with the prefix “`system_`” can be used as substitute for a CSS color.

```
squared.settings.resourceSystemColors = {
  "system_accent1_0": "white", // Converted to ARGB
  "system_accent1_50": "#FFFFFF", // Will override "system_accent1_0"
  "system_accent1_100": ["white", 0.75], // Different color with opacity
  "system_accent1_200": "hwb(12 50% 0%)", // rgb + hsl
  "system_accent1_300": squared.lib.color.parseColor("#000", 1)
};
```

Note: Android 12 *API 31* is required when using accent colors.¹

Listing 35: `?attr/color*`

```
squared.settings.preferMaterialDesign = "MaterialComponents"; // Required
squared.settings.resourceSystemColors = {
  "system_accent1_0": "white", // Invalid
  "colorPrimary": "white",
  "colorSecondary": "rgb(255 0 0)"
};
```

When any element uses a CSS color it will be substituted with the same value in the global system color map. This behavior can be disabled locally using dataset attributes.

¹ <https://source.android.com/docs/core/display/dynamic-color>

Listing 36: Active (highlight)

```
<div data-android-system-color-children="false">
  <span>Item 1</span>
  <span data-android-system-color="true">Item 2</span>
  <div>
    <span>Item 3</span>
  </div>
  <div>
    <span data-android-system-color="false">Item 4</span>
  </div>
  <div data-android-system-color="true">
    <span data-android-system-color="inherit">Item 5</span>
  </div>
  <div>
    <span data-android-system-color="inherit">Item 6</span>
  </div>
</div>
```

Redirecting Output

Sometimes it is necessary to extract elements and append them into other containers for it to look identical on the Android device. Usually it is when you are using an [App widget](#) and never with the standard layouts.

```
<div>
  <span>Item 1</span>
  <span data-android-target="location">Item 2</span>
  <span data-android-target="location" data-android-target-index="1">Item 3</span>
</div>
<ul id="location">
  <li>Item 4</li>
  <li>Item 5</li>
  <!-- span -->
</ul>
```

```
<LinearLayout>
  <TextView>Item 1</TextView>
</LinearLayout>
<LinearLayout>
  <TextView>Item 4</TextView>
  <TextView>Item 3</TextView>
  <TextView>Item 5</TextView>
  <TextView>Item 2</TextView>
</LinearLayout>
```

Note: Using **target** into a *ConstraintLayout* or *RelativeLayout* container will not include automatic positioning. Redirection will fail if the *target location* is not a block/container element.

6.1.14 Settings

These are the latest server default settings for the *android* framework.

Listing 37: squared.json^{Page 77, 1}

```
{
  "apiVersion": "1.4.0",
  "document": {
    "android": {
      "handler": "@pi-r/android",
      "namespace": "",
      "extensions": [
        "@pi-r/android/extensions/app/manifest",
        "@pi-r/android/extensions/gradle/settings",
        "@pi-r/android/extensions/gradle/dependencies",
        "@pi-r/android/extensions/task"
      ],
      "versions": {
        "org.jetbrains.kotlin:kotlin-stdlib": "1.9.23",
        "kotlinCompilerExtensionVersion": "1.5.12",
        "buildToolsVersion": ""
      },
      "settings": {
        "broadcast_id": "",
        "users": {
          "username": {
            "extensions": null
          }
        },
        "extensions": {
          "task": {
            "exec": {
              "uid": "",
              "gid": ""
            },
            "command": ""
          }
        },
        "language": {
          "gradle": "kotlin"
        },
        "directory": {
          "template": ""
        }
      },
      "permission": {}
    }
  }
}
```

¹ <https://www.unpkg.com/squared-express/dist/squared.json>

Interface

Listing 38: View Module

```
interface DocumentModule {
  handler: "@pi-r/android";
  extensions?: string[];
  versions?: StringMap;
  settings?: {
    broadcast_id?: string | string[];
    users?: Record<string, Record<string, unknown>>;
    extensions?: {
      task?: {
        exec?: {
          uid?: number | string;
          gid?: number | string;
        };
        command?: string;
      };
    };
    language?: {
      gradle?: "java" | "kotlin" | "java+kotlin";
    };
    directory?: {
      template?: string;
    };
  };
  permission: PermittedDirectories;
}
```

6.2 Chrome

General usage instructions for *chrome* are not found on this website except in cases where it overlaps with the *android* framework.

Tip: *chrome* is the reference framework for the [E-mc](#) documentation.

6.3 VDOM

6.3.1 Interface

```
class Container<T> implements Iterable<T> {
  children: T[];
  [Symbol.iterator](): IterableIterator<T>;
  item(index: number, value?: T): T | undefined;
  add(item: T): this;
  addAll(list: T[] | Container): this;
```

(continues on next page)

(continued from previous page)

```

    remove(item: T): T | undefined;
    removeAll(list: T[] | Container): T[];
    retainAs(list: T[]): this;
    each(predicate: IteratorPredicate<T, void>, options?: ContainerEachOptions): this;
    every(predicate: IteratorPredicate<T, unknown>, options?: ContainerRangeOptions):
↳ boolean;
    removeIf(predicate: IteratorPredicate<T, unknown>, options?: ContainerRemoveIfOptions
↳ <T> | IteratorPredicate<T, boolean>): T[];
    find(predicate: IteratorPredicate<T, unknown>, options?: ContainerFindOptions<T>): T
↳ | undefined;
    includes(item: T, cascade: boolean): boolean;
    includes(item: T, options?: ContainerIncludesOptions<T>): boolean;
    cascade(predicate?: IteratorPredicate<T, unknown>, options?: ContainerCascadeOptions
↳ <T> | IteratorPredicate<T, boolean>): T[];
    map(predicate: IteratorPredicate<T, unknown>): unknown[];
    contains(...items: T[]): boolean;
    clear(): this;
    isEmpty(): boolean;
    size(): number;
    toArray(): T[];
    constructor(children?: T[]);
}

class Node extends Container<Node> {
    static readonly TEXT_STYLE: string[];
    static boxValueOf(node: Node, attr: BoxModelAttr, options?: CssUnitOptions): number;
    static sanitizeCss(element: HTMLElement, input: CssStyleMap, writingMode?: string,
↳ output?: CssStyleMap): CssStyleMap;
    readonly id: number;
    readonly sessionId: string;
    setParent(parent: Node | null, depth?: number, index?: number): void;
    syncWith(sessionId?: string, cache?: boolean): boolean;
    saveAsInitial(): void;
    elementAttr(attr: string, value: unknown): unknown;
    data(name: string | symbol, attr: string | symbol, value?: unknown, overwrite?:
↳ boolean): unknown;
    unsetCache(...attrs: (CssStyleAttr | keyof CacheValue)[]): void;
    unsetState(...attrs: (keyof CacheState<Node>)) : void;
    ascend(options?: AscendParameterOptions<Node, NodeParentAttr>, attr?: boolean |
↳ string | ((item: Node) => boolean), error?: (item: Node) => boolean): Node[];
    ascendOne(options?: AscendParameterOptions<Node, NodeParentAttr>, attr?: boolean |
↳ string | ((item: Node) => boolean), error?: (item: Node) => boolean): Node | undefined;
    descend(options?: DescendParameterOptions<Node>): Node[];
    descendOne(options?: DescendParameterOptions<Node>): Node | undefined;
    intersectX(rect: BoxRectDimension, options?: CoordsXYOptions): boolean;
    intersectY(rect: BoxRectDimension, options?: CoordsXYOptions): boolean;
    withinX(rect: BoxRectDimension, options?: OffsetXYOptions): boolean;
    withinY(rect: BoxRectDimension, options?: OffsetXYOptions): boolean;
    outsideX(rect: BoxRectDimension, options?: OffsetXYOptions): boolean;
    outsideY(rect: BoxRectDimension, options?: OffsetXYOptions): boolean;
    as(target: typeof Element): InstanceType<typeof Element> | null;
    css(attr: CssStyleAttr, value?: string, cache?: boolean): string;

```

(continues on next page)

(continued from previous page)

```

cssInitial(attr: CssStyleAttr, options?: CssInitialOptions): string;
cssAny(attr: CssStyleAttr, values: string[], options?: CssAnyOptions): boolean;
cssAscend(attr: CssStyleAttr, options?: CssAscendOptions): string;
cssSort(attr: CssStyleAttr, options?: CssSortOptions): Node[];
cssSpecificity(attr: CssStyleAttr): Specificity | undefined;
cssParent(attr: CssStyleAttr, value?: string, cache?: boolean): string;
cssUnit(attr: CssStyleAttr, options?: CssUnitOptions): number;
cssAsTuple(...attrs: CssStyleAttr[]): string[];
cssAsObject(...attrs: CssStyleAttr[]): CssStyleMap;
cssApply(values: CssStyleMap, overwrite?: boolean, cache?: boolean): this;
cssCopy(node: Node, ...attrs: CssStyleAttr[]): void;
cssCopyIfEmpty(node: Node, ...attrs: CssStyleAttr[]): void;
cssTry(attr: CssStyleAttr, value: string, callback?: (this: Node, ...args:
↳ unknown[]) => void): boolean;
cssTryAll(attrs: CssStyleMap, callback?: (this: Node, ...args: unknown[]) => void):
↳ CssStyleMap | boolean;
cssFinally(attrs: CssStyleAttr | CssStyleMap): void;
parseUnit(value: unknown, options?: NodeParseUnitOptions): number;
withDisplay(type: boolean, ...values: string[]): boolean;
withDisplay(...values: string[]): boolean;
withLayout(outside: "block" | "inline" | number, type: number, ...values: string[]):
↳ boolean;
withLayout(outside: "block" | "inline" | number, ...values: string[]): boolean;
parseColor(value: string, opacity: number): ColorRGB | null;
parseColor(value: string, options?: NodeParseColorOptions): ColorRGB | null;
convertUnit(value: unknown, options: NodeConvertUnitOptions): string;
convertUnit(value: unknown, unit?: string, options?: NodeConvertUnitOptions): string;
has(attr: CssStyleAttr, options?: HasOptions): boolean;
hasUnit(attr: CssStyleAttr, percent: boolean): boolean;
hasUnit(attr: CssStyleAttr, options?: HasUnitOptions | boolean): boolean;
toInt(attr: CssStyleAttr, options: CssInitialOptions): number;
toInt(attr: CssStyleAttr, fallback?: number, options?: CssInitialOptions): number;
toFloat(attr: CssStyleAttr, options: CssInitialOptions): number;
toFloat(attr: CssStyleAttr, fallback?: number, options?: CssInitialOptions): number;
toElementInt(attr: string, fallback?: number): number;
toElementFloat(attr: string, fallback?: number): number;
toElementBoolean(attr: string, fallback?: boolean): boolean;
toElementString(attr: string, fallback?: string): string;
setBounds(cache?: boolean): BoxRectDimension | null;
resetBounds(recalibrate?: boolean): void;
getContainerSize(options?: NodeUnitOptions): number;
flex(attr: "inline" | "row" | "column" | "reverse" | "wrap" | "wrapReverse", parent?
↳ : boolean | Node): boolean;
flex(attr: "alignContent" | "justifyContent" | "basis" | "alignSelf" | "justifySelf",
↳ parent?: boolean | Node): string;
flex(attr: "grow" | "shrink" | "order", parent?: boolean | Node): number;
flex(attr: string, parent?: boolean | Node): boolean | number | string;
min(attr: string, options?: MinMaxOptions): Node;
max(attr: string, options?: MinMaxOptions): Node;
querySelector(value: string): Node | null;
querySelectorAll(value: string, queryMap?: Node[], queryRoot?: HTMLElement | null):
↳ Node[];

```

(continues on next page)

(continued from previous page)

```

    ancestors(value?: string | AscendParameterOptions<Node, NodeParentAttr>, options?:
↳ AscendParameterOptions<Node, NodeParentAttr>): Node[];
    ancestorsOne(value?: string | AscendParameterOptions<Node, NodeParentAttr>, options?
↳ : AscendParameterOptions<Node, NodeParentAttr>): Node | undefined;
    descendants(value?: string | DescendParameterOptions<Node>, options?:
↳ DescendParameterOptions<Node>): Node[];
    descendantsOne(value?: string | DescendParameterOptions<Node>, options?:
↳ DescendParameterOptions<Node>): Node | undefined;
    siblings(value?: string | SiblingsParameterOptions<Node>, options?:
↳ SiblingsParameterOptions<Node>): Node[];
    siblingsOne(value?: string | SiblingsParameterOptions<Node>, options?:
↳ SiblingsParameterOptions<Node>): Node | undefined;
    boxOf(attr: keyof (BoxRect & Dimension)): number;
    valueOf(attr: CssStyleAttr, options?: CssInitialOptions): string;
    get documentRoot(): boolean;
    get parent(): Node | null;
    get shadowRoot(): boolean;
    get shadowHost(): ShadowRoot | null;
    get depth(): number;
    get childIndex(): number;
    get naturalChildren(): Node[];
    get naturalElements(): Node[];
    get dir(): string;
    get textBounds(): BoxRectDimension | null;
    get box(): BoxRectDimension;
    get bounds(): BoxRectDimension;
    get linear(): BoxRectDimension;
    get element(): Element | null;
    get elementId(): string;
    get tagName(): string;
    get naturalChild(): boolean;
    get naturalElement(): boolean;
    get parentElement(): Element | null;
    get htmlElement(): boolean;
    get styleElement(): boolean;
    get imageElement(): boolean;
    get svgElement(): boolean;
    get flexElement(): boolean;
    get gridElement(): boolean;
    get textElement(): boolean;
    get tableElement(): boolean;
    get inputElement(): boolean;
    get buttonElement(): boolean;
    get mathElement(): boolean;
    get voidElement(): boolean;
    get pseudoElement(): boolean;
    get pseudoElt(): PseudoElt | "";
    get documentBody(): boolean;
    get dataset(): DOMStringMap;
    get centerAligned(): boolean;
    get rightAligned(): boolean;
    get bottomAligned(): boolean;

```

(continues on next page)

(continued from previous page)

```

get width(): number;
get height(): number;
get hasWidth(): boolean;
get hasHeight(): boolean;
get lineHeight(): number;
get display(): string;
get positionStatic(): boolean;
get positionRelative(): boolean;
get positionFixed(): boolean;
get top(): number;
get right(): number;
get bottom(): number;
get left(): number;
get borderTopWidth(): number;
get borderRightWidth(): number;
get borderBottomWidth(): number;
get borderLeftWidth(): number;
get marginTop(): number;
get marginRight(): number;
get marginBottom(): number;
get marginLeft(): number;
get paddingTop(): number;
get paddingRight(): number;
get paddingBottom(): number;
get paddingLeft(): number;
get outlineWidth(): number;
get inline(): boolean;
get inlineStatic(): boolean;
get inlineText(): boolean;
get block(): boolean;
get blockStatic(): boolean;
get plainText(): boolean;
get textContent(): string;
get lineBreak(): boolean;
get pageFlow(): boolean;
get autoMargin(): AutoMargin;
get floating(): boolean;
get float(): FloatDirectionAttr;
get floatClear(): ClearDirectionAttr;
get baseline(): boolean;
get multiline(): boolean;
get contentBox(): boolean;
get contentBoxWidth(): number;
get contentBoxHeight(): number;
get borderBoxElement(): boolean;
get flexdata(): FlexData;
get flexbox(): FlexBox;
get zIndex(): number;
get opacity(): number;
get backgroundColor(): string;
get backgroundImage(): string;
get visibleStyle(): VisibleStyle;

```

(continues on next page)

(continued from previous page)

```

get fontSize(): number;
get verticalAlign(): number;
get actualParent(): Node | null;
get absoluteParent(): Node | null;
get wrapperOf(): Node | null;
get actualWidth(): number;
get actualHeight(): number;
get actualDimension(): Dimension;
get containerDimension(): Readonly<ContainerDimension>[] | null;
get containerHeight(): boolean;
get percentWidth(): number;
get percentHeight(): number;
get aspectRatio(): [number, number] | null;
get firstChild(): Node | null;
get lastChild(): Node | null;
get firstElementChild(): Node | null;
get lastElementChild(): Node | null;
get previousSibling(): Node | null;
get nextSibling(): Node | null;
get previousElementSibling(): Node | null;
get nextElementSibling(): Node | null;
get attributes(): StringMap;
get checked(): boolean | null;
get boundingClientRect(): DOMRect | null;
get preserveWhiteSpace(): boolean;
get style(): CSSStyleDeclaration;
get cssStyle(): CssStyleMap;
get textStyle(): CssStyleMap;
get writingMode(): string;
get elementData(): ElementData | null;
get initial(): InitialData<Node>;
constructor(id: number, sessionId?: string, element?: Element, children?: Node[]);
}

```

Added in version 5.2.1:

- *Node* method **as** for element casting was created.

Added in version 5.2.0:

- *Node* method **withDisplay** for group display values was created.
- *Node* method **withLayout** for display outside and inside values was created.
- *Node* method **parseColor** for color scheme detection was created.
- *Node* method **flex** for parsed Flexbox values was created.
- *Node* property getter **mathElement** for MathML elements was created.
- *Container* methods **removeIf** | **cascade** argument **options** with *IteratorPredicate* was amended.

Deprecated since version 5.2.0:

- *Node* property getter **flexdata** as *Node.flex(attr, parent)* is pending removal.

Listing 39: squared.lib.js

```
class Container {
  addAt(index: number, ...items: T[]): this;
  removeAt(index: number): T | undefined;
  sortBy(...attrs: [...string[], boolean?]): this;
  iterator(): ListIterator<T>;
}
```

6.4 Build Options

squared

- *FileActionOptions*

squared-express

- *DocumentOutput*
- *RequestData*

6.4.1 Interface

```
interface FileActionOptions {
  pid?: number;
  fetchMode?: RequestMode;
  timeout?: number;
  config?: boolean | string | FileActionConfig;
  incrementalMap?: {
    pathname?: IncrementalData;
    extension?: IncrementalData;
    mime?: IncrementalData;
    overwrite?: boolean;
  };
  outgoing?: OutgoingHeaders;
  exclusions?: Exclusions | (string | RegExp)[];
  broadcast?: BroadcastSocket | BroadcastMessageCallback;
  throwErrors?: boolean;
  filter?: (item: FileAsset, index: number, array: FileAsset[]) => unknown;
}

interface RequestData {
  projectId?: string;
  assets?: FileAsset[];
  dataSource?: DataSource[];
  document?: string[];
  task?: string[];
  modules?: string[];
  update?: WatchInterval;
  incremental?: boolean | "none" | "staging" | "etag" | "exists";
  checksum?: string | boolean | 1 | ChecksumOutput;
}
```

(continues on next page)

(continued from previous page)

```

imports?: StringMap;
headers?: OutgoingHeaders;
watch?: boolean;
cache?: boolean | PlainObject;
log?: boolean | string | string[] | LogOptions;
baseUrl?: string;
priority?: number;
broadcastId?: string | string[];
error?: { abort?: boolean | string | string[]; fatal?: boolean };
ignoreExtensions?: boolean | string | string[];
}

```

```

interface FileCopyingOptions extends FileActionOptions {
  pathname?: string;
  emptyDir?: boolean;
  modified?: boolean;
}

interface FileArchivingOptions extends FileActionOptions {
  filename?: string;
  format?: string;
  copyTo?: string;
}

```

6.4.2 Android

Interface

```

interface DocumentOutput {
  targetAPI?: number | string;
  manifest?: ManifestData;
  namespace?: string;
  profileable?: boolean | string | string[];
  dependencies?: string[];
  dependencyScopes?: boolean | 1 | DependencyScopes | "snapshot" | (DependencyScopes |
↪ "snapshot")[];
  directories?: ControllerSettingsDirectoryUI;
  projectName?: string;
  mainParentDir?: string;
  mainSrcDir?: string;
  mainActivityFile?: string;
  javaVersion?: number | string;
  versionName?: string;
  versionCode?: number;
  dataBinding?: boolean;
  commands?: string | string[] | (string | string[])[];
  extensionData?: Record<string, PlainObject>;
  updateXmlOnly?: boolean;
}

```

Added in version 5.2.0:

- `dependencyScopes` with the “snapshot” number value “1” for all scopes.

Example usage

```
squared.saveAs("android.zip", {
    targetAPI: 32, // Override settings.targetAPI
    targetAPI: "Tiramisu",
    manifest: {
        package: "com.example.demo", // <manifest package="com.example.demo">
        application: {
            label: "app_name",
            supportsRtl: true,
            theme: "AppTheme"
        }
    },
    namespace: "com.example.demo", // android.defaultConfig.applicationId (app/build.
    ↪ gradle)
    profileable: true, // <profileable android:enabled="[false|true]" />
    profileable: "debug", // android.buildTypes.release.signingConfig = signingConfigs.
    ↪ debug
    profileable: "--warn-manifest-validation", // aaptOptions.additionalParameters (--
    ↪ prefix)
    profileable: ["release", "--warn-manifest-validation", "--no-version-vectors"], // ↵
    ↪ signingConfig + additionalParameters (multiple --args)
    dependencies: ["androidx.appcompat:appcompat:1.6.0"],
    dependencyScopes: true, // All first-level dependencies
    dependencyScopes: "compile", // implementation="compile" | compileOnly="provided" | ↵
    ↪ runtimeOnly="runtime" | testImplementation="test"
    dependencyScopes: ["compile", "runtime"],
    dependencyScopes: "snapshot", // Use latest published release
    dependencyScopes: 1, // true + "snapshot"
    dependencyScopes: ["snapshot", "compile"],
    directories: {
        layout: "/path/to/res/layout",
        string: "/path/to/res/values"
    },
    projectName: "Example Project", // rootProject.name (settings.gradle)
    mainParentDir: "app", // Override settings.outputDirectory
    mainSrcDir: "src/main",
    mainActivityFile: "MainActivity.java", // "MainActivity.*" | "/path/user/project/
    ↪ MainActivity.java" | "app/path/MainActivity.java"
    javaVersion: 1.8, // JavaVersion.VERSION_1_8
    javaVersion: 11, // JavaVersion.VERSION_11
    versionName: "1.0",
    versionCode: 1,
    dataBinding: true, // android.buildFeatures.dataBinding
    commands: "build", // gradlew build
    commands: ["test", "deploy"], // gradlew test deploy
    commands: ["lint", ["test", "--rerun-tasks"]], // gradlew lint && gradlew test --rerun-
    ↪ tasks
    updateXmlOnly: true // Copy only auto-generated content
});
```

Listing 40: With assets

```
squared.saveAs("android.zip", {
  projectId: "project-1",
  assets: [
    {
      pathname: "app/src/main/res/drawable",
      filename: "ic_launcher_background.xml",
      uri: "http://localhost:3000/common/images/ic_launcher_background.xml"
    },
    {
      pathname: "app/src/main/res/drawable-v24",
      filename: "ic_launcher_foreground.xml",
      uri: "http://localhost:3000/common/images/ic_launcher_foreground.xml"
    }
  ]
});
```

6.4.3 Chrome

Interface

```
interface FileActionOptions {
  baseHref?: URL;
  saveAs?: {
    html?: SaveAsOptions;
    script?: SaveAsOptions;
    link?: SaveAsOptions;
    image?: SaveAsOptions;
    font?: SaveAsOptions;
  };
  downloadOnly?: boolean;
  excluding?: HTMLElement[];
  observe?: true | MutationCallback;
  preserveCrossOrigin?: boolean | URLData;
  addResourceHints?: boolean | ResourceHintType | ResourceHintType[];
  retainUsedStyles?: (string | RegExp)[] | UsedStylesData;
  removeInlineStyles?: boolean;
  removeUnusedClasses?: boolean;
  removeUnusedPseudoClasses?: boolean;
  removeUnusedVariables?: boolean;
  removeUnusedFontFace?: boolean;
  removeUnusedKeyframes?: boolean;
  removeUnusedMedia?: boolean;
  removeUnusedSupports?: boolean;
  removeUnusedContainer?: boolean;
  removeUnusedScope?: boolean;
}

interface DocumentOutput {
  productionRelease?: boolean | string;
}
```

(continues on next page)

(continued from previous page)

```

useOriginalHtmlPage?: boolean | string;
useUnsafeHtmlReplace?: boolean;
useSessionCache?: boolean;
stripCommentsAndCDATA?: boolean | string;
normalizeHtmlOutput?: boolean | string;
escapeReservedCharacters?: boolean;
webBundle?: {
  rootDirAlias?: string;
  baseUrl?: string;
  primaryUrl?: string;
  copyTo?: string;
  rewriteHtmlPage?: boolean | string;
  excludeHtmlPage?: boolean;
  includeScopes?: string[];
  excludeScopes?: string[];
};
templateMap?: TemplateMap;
userAgentData?: UserAgentData;
/* Auto-generated from "removeUnused" */
usedVariables?: string[]; // User supplied prepended
usedFontFace?: string[];
usedKeyframes?: string[];
unusedStyles?: string[];
unusedMedia?: string[];
unusedSupports?: string[];
unusedContainer?: string[];
unusedScope?: string[];
unusedAtRules?: UnusedAtRule[];
}

```

Added in version 5.2.0:

- *FileActionOptions* property **removeUnusedScope** was created.

Example usage

```

squared.copyTo("/path/project", {
  cache: {
    transform: false, // Not recommended when using watch
    transform: true, // "etag" (not bundled) + string comparison by URL (single page)
    transform: "etag", // request.cache OR request.buffer.expires (required)
    transform: "md5" | "sha1" | "sha224" | "sha256" | "sha384" | "sha512", // Multi-
    ↪ [user/page] + Inline content (includes "etag")
    transform: { expires: "2h" }, // Expires in 2 hrs since creation
    transform: { expires: "1h", renew: true }, // Expires from 1 hr of last time accessed
    transform: { algorithm: "md5" /* etag */, expires: "2h", limit: "5mb" }, // Set
    ↪ expiration and content size limit
    transform: { exclude: { html: "*", js: ["bundle-es6"] } }, // Format names per type
    transform: { include: { css: "*", js: ["bundle"] } }
  },
  checksum: true, // sha256 + recursive

```

(continues on next page)

(continued from previous page)

```

checksum: 1, // sha256 + recursive = 1
checksum: "sha512", // checksum.sha512
checksum: "filename.sha384", // sha384
checksum: {
  algorithm: "md5", // Default is "sha256"
  digest: "base64", // Default is "hex"
  filename: "checksum.crc", // Default is "checksum" + algorithm
  recursive: true, // Default is "false"
  recursive: 1, // Ignore nested checksum files
  include: "**/*.png", // Has precedence
  exclude: ["**/*.js", "**/*.css"]
},
imports: {
  "http://localhost:3000/build/": "./build", // Starts with "http"
  "http://localhost:3000/dist/chrome.framework.js": "/path/project/build/framework/
↳ chrome/src/main.js" // Full file path
},
webBundle: { // Chromium
  baseUrl: "http://hostname/dir/", // Resolves to current host and directory
  rewriteHtmlPage: true | "index.html", // Hide or rename main page
  excludeHtmlPage: true, // Exclude HTML page from WBN archive
  excludeTransforms: true, // Exclude transformed files not used in HTML page
  includeScopes: ["**/*.css"], // http://localhost:3000/dir/**/*.css (hides
↳ "excludeTransforms" + "excludeScopes")
  excludeScopes: ["**/*.js"], // http://localhost:3000/**/*.js
  copyTo: "/path/project", // Copy archive (absolute + permission)
  rootDirAlias: "__serverroot__" // Internal
},
baseHref: "http://hostname/prod/example.html", // Additional hostname to use for
↳ parsing local files
retainUsedStyles: [/^a:[a-z]/i, "--property-name"],
downloadOnly: true, // Do not transform HTML and CSS files
excluding: Array.from(document.querySelectorAll("video, audio")) // Elements to remove
↳ from HTML
});

```

See also:

E-mc / Build Options

RELEASE NOTES

7.1 5.3.0

2024-08-06

7.1.1 BREAKING

Listing 1: squared.base

```
class NodeUI {  
    get textIndent(): number; // Detection for "each-line" and "hanging"  
}
```

7.1.2 ADDED

Listing 2: squared.base

```
interface ControllerSettingsStyleUI {  
    inputFontSize?: string;  
    buttonFontSize?: string;  
    textareaFontSize?: string;  
    selectFontSize?: string;  
    h1FontSize?: string;  
    h2FontSize?: string;  
    h3FontSize?: string;  
    h4FontSize?: string;  
    h5FontSize?: string;  
    h6FontSize?: string;  
    rtFontSize?: string;  
}  
  
interface HasOptions {  
    inherit?: boolean;  
}  
  
class NodeUI {  
    get afterLineBreak(): boolean;
```

(continues on next page)

(continued from previous page)

```
    get wordSpacing(): number;
}
```

Listing 3: squared.lib

```
/* dom */
function causesLineBreak(element: Element): boolean;
```

7.1.3 MODIFIED

Listing 4: squared

```
function toString(projectId?: string): string; // projectId
```

Listing 5: squared.base

```
interface HasOptions {
    type?: number; // CSS_UNIT.DECIMAL | CSS_UNIT.INTEGER
}
```

7.1.4 DEPRECATED

Listing 6: squared.base

```
interface ControllerSettingsStyleUI {
    formFontSize: string;
}

class ApplicationUI {
    get layouts(): FileAsset[]; // ApplicationUI.getProject().layouts
}
```


7.2 5.2.0

7.2.1 5.2.1

2024-05-03

ADDED

Listing 7: squared.base

```
interface HasUnitOptions {
    inherit?: boolean;
    ignoreZero?: boolean;
}

class Node {
    as(target: typeof Element): InstanceType<typeof Element> | null;
}

class NodeUI {
    get textMetrics(): TextMetrics | null;
}
```

Listing 8: android.extensions.resource

```
interface ResourceSvgOptions {
    dependencyVectorDrawable: "animated" | "seekable";
}
```

2024-04-29

7.2.2 BREAKING

Listing 9: squared.base

```
class Node {
    toElementInt(attr: string, fallback?: number): number; // Uses Math.trunc (rounds to ⬇
    ↪ 0)
}
```

Listing 10: squared.lib.client

```
interface UserAgentData {
    browser: Chrome = 1;
    browser: Edge = 8;

    browser: Edge = 1 & 8; // getUserAgentData
}
```

Listing 11: squared.svg

```

class Svg {
  constructor(element: SVGSVGElement, documentRoot?: boolean); // documentRoot not
  ↳ "true" by default (parentElement instanceof HTMLElement)
}

class SvgAnimateTransform {
  get attributeName(): string; // "transform" for <animateTransform>
  get attributeName(): string; // "translate" | "scale" | "rotate" | "skewX" | "skewY"
  ↳ for CSS animation
}

class SvgAnimateMotion {
  set path(value); // Will refit path automatically
}

```

Listing 12: types/base/squared.d.ts

```

/* 5.1 */

type UserSettingMethod<T, U, V> = <W = unknown>(id: number | string | undefined, name:
  ↳ keyof U, fallback?: W) => V extends true ? W : W | undefined;

class Application<T, U, V, W extends UserResourceSettings> {
  getUserSetting: UserSettingMethod<T, W, true>; // Without undefined
  getCustomSetting: UserSettingMethod<T, W, false>; // With undefined
}

class Extension<T, K, V> {
  readonly project: ProjectMap<K, V>;
  getUserSetting: UserSettingMethod<T, PlainObject, false>;
  getCustomSetting: UserSettingMethod<T, PlainObject, false>;
}

/* 5.2 */

type UserSettingMethod<T, U> = <V extends keyof U>(id: number | string | undefined,
  ↳ name: V, fallback?: NoInfer<U[V]>) => U[V] | undefined; // TypeScript 5.4

class Application<T, U, V, W extends UserResourceSettings> {
  getUserSetting: UserSettingMethod<T, W>; // With undefined
  getCustomSetting: UserSettingMethod<T, W>; // With undefined
}

class Extension<T, U extends PlainObject, K, V> {
  readonly options: U;
  readonly project: ProjectMap<K, V>;
  getUserSetting: UserSettingMethod<T, U>;
  getCustomSetting: UserSettingMethod<T, U>;
}

```

7.2.3 ADDED

Listing 13: squared

```
interface Point {
  z?: number;
}
```

Listing 14: squared.base

```
interface UserSettings {
  adaptStyleMap: boolean;
}

interface UserResourceSettingsUI {
  useShapeGeometryBox?: boolean;
}

interface ControllerSettingsUI {
  baseStyle?: ControllerSettingsBaseStyleUI;
}

interface ControllerSettingsBaseStyleUI {
  math?: CssStyleMap;
}

interface Flexbox {
  enabled: boolean;
}

interface NodeParseColorOptions {
  opacity?: number;
  ignoreWhite?: boolean;
  ignoreBlack?: boolean;
  ignoreTransparent?: boolean;
}

interface RequestData {
  log?: { showProgress?: boolean };
}

class ResourceUI {
  static findStoredAsset(resourceId: number, type: string, name: string, remove?: boolean): unknown;
}

class Node {
  parseColor(value: string, opacity: number): ColorRGB | null;
  parseColor(value: string, options?: NodeParseColorOptions): ColorRGB | null;
  withDisplay(type: number, ...values: string[]): boolean;
  withDisplay(...values: string[]): boolean;
  withLayout(outside: "block" | "inline" | number, type: number, ...values: string[]): boolean;
}
```

(continues on next page)

(continued from previous page)

```

    withLayout(outside: "block" | "inline" | number, ...values: string[]): boolean;
    flex(attr: "inline" | "row" | "column" | "reverse" | "wrap" | "wrapReverse", parent?
↳: boolean | Node): boolean;
    flex(attr: "alignContent" | "justifyContent" | "basis" | "alignSelf" | "justifySelf",
↳ parent?: boolean | Node): string;
    flex(attr: "grow" | "shrink" | "order", parent?: boolean | Node): number;
    get mathElement(): boolean;
}

class NodeUI {
    get tagDisplay(): string;
    get alignContent(): "start" | "end" | "center" | "baseline" | "";
}

```

Listing 15: squared.base.lib

```

/* constant */

const enum TRANSFORM_METHOD {
    NONE = 0,
    TRANSLATE_X = 1,
    ROTATE_X = 2,
    SCALE_X = 4,
    SKEW_X = 8,
    TRANSLATE_Y = 16,
    ROTATE_Y = 32,
    SCALE_Y = 64,
    SKEW_Y = 128,
    TRANSLATE_Z = 256,
    ROTATE_Z = 512,
    SCALE_Z = 1024,
    TRANSLATE_2D = TRANSLATE_X | TRANSLATE_Y,
    SCALE_2D = SCALE_X | SCALE_Y,
    ROTATE_2D = ROTATE_X | ROTATE_Y,
    TRANSLATE = TRANSLATE_2D | TRANSLATE_Z,
    ROTATE = ROTATE_2D | ROTATE_Z,
    SCALE = SCALE_2D | SCALE_Z,
    SKEW = SKEW_X | SKEW_Y,
    ALL = TRANSLATE | ROTATE | SCALE | SKEW,
    ALL_2D = TRANSLATE_2D | ROTATE | SCALE_2D | SKEW
}

/* CSS */

function parseBorderRadius(value: string, options?: ParseBorderRadiusOptions): [number,
↳number, number, number][] | null;

/* util */

function addMimeType(types: StringMap): void;
function addMimeType(name: string, value: string): void;

```

Listing 16: squared.lib

```

/* client */

interface UserAgentData {
    name: string;
    majorVersion: number;
    mobile: boolean;
}

function getUserAgentFeatures(): UserAgentFeatures;

/* color */

function isColor(value: unknown): value is ColorRGB;

/* constant */

const enum CSS_DISPLAY {
    INLINE = 1,
    INLINE_START = 2,
    FLOW = 4,
    FLOW_START = 8,
    TABLE = 16,
    TABLE_START = 32,
    RUBY = 64,
    RUBY_START = 128
}

const enum CSS_LAYOUT {
    NONE = 0,
    BLOCK = 1,
    INLINE = 2,
    FLOW = 4,
    FLOW_ROOT = 8,
    TABLE = 16,
    FLEX = 32,
    GRID = 64,
    RUBY = 128,
    MATH = 256,
    LIST_ITEM = 512
}

/* container */

interface ContainerRangeOptions {
    isEmpty?: boolean
}

/* CSS */

function measureText(element: HTMLElement, value: string, options?: MeasureTextOptions):

```

(continues on next page)

(continued from previous page)

```

↪number;

/* math */

function oppositeAngle(angle: number, x: number, y: number, center?: Point | null, ↪
↪relative?: boolean): number;
function truncateAngle(angle: number): number;
function distance(start: Point, end: Point): number;
function containPoint(angle: number, x: number, y: number, rect: BoxRect | DOMRect): ↪
↪Point;
function hasPoint(x: number, y: number, rect: BoxRect | DOMRect, offset?: number): ↪
↪boolean;

/* regex */

const CSS: {
    LIGHT_DARK: RegExp;
    CUSTOM_ELEMENT: RegExp;
};

/* util */

function selectSet(value: string, delimiter: string | RegExp, ...items: string[]): Set
↪<string>;

```

Listing 17: squared.svg

```

type SvgGeometryBoxValue = "border-box" | "fill-box" | "stroke-box" | "view-box" |
↪"margin-box" | "padding-box" | "content-box";

interface SvgPaint {
    clipRule: string;
    getBoundingBox(paths?: string[] | DOMRect, geometryBox?: SvgGeometryBoxValue): ↪
↪DOMRect;
    getStrokeWidth(fallback: number): number;
    getStrokeWidth(method?: SvgMathRounding, fallback?: number): number;
}

class SvgBuild {
    static isPaint(object: unknown): object is SvgPaint;
    static drawShapes(element: SVGGraphicsElement, values: ArrayOf<string>, options?: ↪
↪SvgDrawShapesOptions<SvgContainer>): SvgShapePath[];
    static transformPoints(values: SvgPoint[], transforms: SvgTransform[], options?: ↪
↪SvgTransformPointsOptions): SvgPoint[];
}

class SvgContainer {
    findViewBox(element?: SVGSVGElement): SVGSVGElement | SVGSymbolElement | undefined;
    setAspectRatio(container?: Svg | SvgUseSymbol | null, options?: ↪
↪SvgAspectRatioOptions): void;
    refitRect(x1: number, y1: number, width: number, height: number): DOMRect;
}

```

(continues on next page)

(continued from previous page)

```

class SvgPath {
  draw(options?: SvgBuildOptions): string;
  get transformOrigin(): Point | null;
}

class SvgShape {
  get transformOrigin(): Point | null;
}

class SvgUseShape {
  get transformOrigin(): Point | null;
}

class SvgAnimation {
  refitPoint(value: Point | null): void;
  get precision(): number | undefined;
}

class SvgAnimate {
  startValues: StringMap | null;
  readonly replaceType: Map<number, string>;
}

class SvgAnimateTransform {
  setReplaceValues(): void;
}

class SvgAnimateMotion {
  positionData: NumberValue[] | null;
  set anchor(value);
  get anchor(): Point | null;
  set position(value);
  get position(): Point | null;
  get cssPath(): string;
  get rotateOutput(): SvgRotatePoint[] | null;
}

```

Listing 18: squared.svg.lib.util

```

const SVG: {
  geometry(element: Element): element is SVGGeometryElement;
};

const TRANSFORM: {
  parseStyle(element: SVGElement, all?: boolean, type?: number): SvgTransform[] | null;
  fromList(values: SVGTransformList): SvgTransform[];
  nameAsType(value: string): number;
};

function getColorValue(element: SVGElement, options: SvgColorValueOptions): ColorRGB |
↪ null;

```

(continues on next page)

(continued from previous page)

```

function getColorValue(element: SVGElement, value: string, options?:
↳SvgColorValueOptions): ColorRGB | null;
function copyRect(target: DOMRect): SvgRect & BoxRect;
function isObjectBoundingBox(element: SVGElement): element is SVGClipPathElement |
↳SVGPatternElement;
function getBoundingBox(element: SVGGraphicsElement, geometryBox?: SvgGeometryBoxValue):
↳DOMRect;
function getBoundingLength(element: SVGGraphicsElement, value: string, dimension: number
↳| DimensionAttr, precision: number): number;
function getBoundingLength(element: SVGGraphicsElement, value: string, dimension?:
↳number | DimensionAttr, geometryBox?: SvgGeometryBoxValue, precision?: number): number;
function getBoxPosition(element: SVGElement, attr: SvgBoxPositionAttr, value?: string):
↳Point | null;
function formatRotate(value: string, type?: number): string;
function formatScale(value: string, type?: number): string;
function formatTranslate(value: string, type?: number): string;
function isNone(value: unknown): boolean;

```

Listing 19: android.base

```

interface UserResourceSettingsUI {
    resourceSystemColors?: Record<string, string | [string, number] | ColorRGB>;
}

interface LocalSettingsUI {
    useSystemColors: boolean;
}

interface AddColorOptions extends NodeParseColorOptions {
    prefix?: string;
    ignoreSystemColors?: boolean;
}

interface AddThemeOptions extends AddColorOptions { /* Empty */}

class View {
    set useSystemColors(value);
    get useSystemColors(): boolean;
}

```

Listing 20: android.extensions.resource

```

interface ResourceSvgOptions {
    mergeClipPath: boolean;
    mergeVectorDrawable: boolean;
}

```

Listing 21: android.lib.constant

```

const MATERIAL_TAGNAME: {
    IMAGE: string;
    LINE: string;
}

```

(continues on next page)

(continued from previous page)

```
};
const SUPPORT_TAGNAME_X: {
  APPBAR_TOOLBAR: string;
};
const XML_PROLOG: string;
const SYSTEM_COLOR: string[];
const ATTRIBUTE_COLOR: string[];
```

7.2.4 MODIFIED

Listing 22: squared.base

```
class NodeUI {
  extractAttributes(depth?: number, replaceWith?: AnyObject): string; // replaceWith
  getPseudoElement(name: PseudoElt | PseudoStyleElt, attr?: CssStyleAttr): string |
  ↳ CssStyleMap | null; // PseudoStyleElt
  flex(attr: "inline" | "row" | "column" | "reverse" | "wrap" | "wrapReverse", parent?
  ↳: boolean | NodeUI, wrapped?: boolean): boolean; // parent
  flex(attr: "alignContent" | "justifyContent" | "basis" | "alignSelf" | "justifySelf",
  ↳ parent?: boolean | NodeUI, wrapped?: boolean): string;
  flex(attr: "grow" | "shrink" | "order", parent?: boolean | NodeUI, wrapped?:
  ↳ boolean): number;
}
```

Listing 23: squared.lib.base

```
class Container {
  removeIf(predicate: IteratorPredicate<unknown, unknown>, options?:
  ↳ ContainerRemoveIfOptions | IteratorPredicate<unknown, boolean>): T[]; // options
  cascade(predicate?: IteratorPredicate<unknown, unknown>, options?:
  ↳ ContainerCascadeOptions | IteratorPredicate<unknown, boolean>): T[];
}
```

Listing 24: squared.lib.math

```
function triangulate(a: number, b: number, clen: number, contain?: boolean): number; //
  ↳ contain
function hypotenuse(a: number, b: number, subtract?: boolean): number; // subtract
```

Listing 25: squared.lib.regex

```
const CSS: {
  COLOR: RegExp; // flags "gi"
};
```

Listing 26: squared.svg

```
interface SvgPaint {
  geometryBox: SvgGeometryBoxValue;
  setPaint(options: SvgBuildOptions): void; // options
```

(continues on next page)

(continued from previous page)

```

    setPaint(d: string[] | null, options: SvgBuildOptions): void;
}

class SvgBuild {
    static drawRect(rect: BoxRect | DOMRect, options?: SvgDrawRectOptions): string; //↵
    ↪DOMRect | options
    static drawRect(width: number, height: number, x: number, y: number, options:↵
    ↪SvgDrawRectOptions): string; // options
    static toOffsetPath(value: string, rotation?: string, options?:↵
    ↪SvgOffsetPathOptions): SvgOffsetPath[]; // options
    static syncPath(values: SvgPathCommand[], points: SvgPoint[], options?:↵
    ↪SvgSyncPathOptions): SvgPathCommand[]; // options
    static boxRectOf(value: string[] | DOMRect, strokeWidth?: number | string):↵
    ↪BoxRectDimension; // DOMRect | BoxRectDimension
}

class SvgPath {
    static extrapolate(attr: string, value: string, values: string[], precision:↵
    ↪number): string[];
    static extrapolate(attr: string, value: string, values: string[], options?:↵
    ↪SvgPathExtrapolateOptions<SvgShape>): string[]; // options
}

class SvgAnimateMotion {
    rotateData: NumberValue[] | null; // undefined
    set distance(value); // Uses current CSS value except when set explicitly
    get distance(): string;
    set rotate(value); // Same
    get rotate(): string;
}

```

Listing 27: squared.svg.lib.util

```

class TRANSFORM: {
    parse(element: SVGElement, type: number): SvgTransform[] | null;
    parse(element: SVGElement, value?: string, type?: number): SvgTransform[] | null;
};

function containsViewBox(element: SVGElement | null, outer?: boolean): element is↵
    ↪SVGSVGElement | SVGSymbolElement | SVGPatternElement | SVGMarkerElement; // outer

```

Listing 28: android.base

```

interface UserResourceSettingsUI {
    builtInExtensions: [
        "android.delegate.content" // CSS: contain | align-content
    ],
    showAttributes: boolean | Record<string, string | null | (string | null)[]>; //↵
    ↪Record
    baseLayoutAsFragment: boolean | string | string[] | ExtensionFragmentElement; //↵
    ↪ExtensionFragmentElement extends ViewAttribute
}

```

(continues on next page)

(continued from previous page)

```

class Resource {
    static formatOptions(resourceId: number, data: ViewAttribute, options?: ↵
↵AddColorOptions): ViewAttribute; // options
    static addColor(resourceId: number, value: ColorRGB | string, options?: ↵
↵AddColorOptions): string; // options
}

```

7.2.5 DEPRECATED

Listing 29: squared.base

```

class Node {
    get flexdata(): FlexboxLayout; // Node.flex(attr, parent)
}

class NodeUI {
    get flexRow(): boolean; // NodeUI.flex("row")
    get flexColumn(): boolean; // NodeUI.flex("column")
}

```

Listing 30: squared.lib

```

/* client */

interface UserAgentData {
    brand: string;
}

/* CSS */

function measureTextWidth(element: StyleElement, value: string, options?: ↵
↵MeasureTextOptions): number; // measureText

/* supports */

function hasLookbehind(): boolean; // client.getUserAgentFeatures
function hasCascadeLayers(): boolean;
function hasContainerQueries(): boolean;
function hasStyleNesting(): boolean;
function hasStyleSheet(): boolean;
function hasStyleProperty(): boolean;

```

Listing 31: squared.svg

```

class SvgBuild {
    static convertTransforms(transforms: SVGTransformList): SvgTransform[]; // lib.util.
↵TRANSFORM.fromList
    static applyTransforms(transforms: SvgTransform[], points: Point[], origin?: Point | ↵
↵null): SvgPoint[]; // transformPoints

```

(continues on next page)

(continued from previous page)

```
}

class SvgPath {
    static extrapolate(attr: string, value: string, values: string[], transforms?:
↳ SvgTransform[] | null, parent?: SvgShape | null, precision?: number): string[]; //
↳ options
}

class SvgAnimateMotion {
    get rotateValues(): number[] | null; // rotateOutput.angle
}
```

Listing 32: squared.svg.lib.util

```
class TRANSFORM: {
    parse(element: SVGElement, with3d: boolean): SvgTransform[] | null; // type
    parse(element: SVGElement, value?: string, with3d?: boolean): SvgTransform[] | null;
↳ // type
};
```

Listing 33: android.base

```
class Resource {
    static addColor(resourceId: number, value: ColorRGB | string, transparent?:
↳ boolean): string; // options
}
```

7.3 5.1.0

7.3.1 5.1.1

5.1.2

5.1.7

2024-04-02

ADDED

Listing 34: squared.svg

```
interface SvgPaint {
    computed: boolean;
}
```

MODIFIED

Listing 35: squared.svg

```
class SvgBuild {
    static boxRectOf(value: string[], strokeWidth?: number | string): BoxRect; // ↵
    ↪strokeWidth
}

class SvgAnimateMotion {
    set path(value);
    get path(): string;
}
```

2023-12-23

ADDED

Listing 36: squared.lib

```
/* CSS */

function detectUnit(value: string, options?: CalculateOptions): [number, number];
function asUnitOf(value: number): string;
function hasLength(value: string): boolean;
function hasColor(value: string): boolean;
function hasAngle(value: string): boolean;
function hasTime(value: string): boolean;

/* util */

function charStart(value: string, char: string): boolean;
function charEnd(value: string, char: string): boolean;
```

Listing 37: squared.svg

```
interface SvgPaint {
    geometryBox: string;
}

class SvgAnimate {
    get timingValues(): unknown[] | null;
}
```

Listing 38: squared.svg.lib.util

```
function containsViewBox(element: SVGElement): element is SvgViewBoxElement;
```

MODIFIED

Listing 39: android.lib.constant

```
const enum DEPENDENCY_TYPE {  
    IMPLEMENTATION = 0,  
    API = 1,  
    COMPILE_ONLY = 2,  
    COMPILE_ONLY_API = 3,  
    RUNTIME_ONLY = 4,  
    TEST_IMPLEMENTATION = 5,  
    TEST_COMPILE_ONLY = 6,  
    TEST_RUNTIME_ONLY = 7,  
    ANDROID_TEST_IMPLEMENTATION = 8,  
    ANDROID_TEST_COMPILE_ONLY = 9,  
    ANDROID_TEST_RUNTIME_ONLY = 10  
}
```

Listing 40: squared.svg.lib.util

```
function findContainer(element: Element, name?: string, ancestors?: WeakMap<HTMLElement,   
↳ ContainerData>): ContainerData | null; // ancestors
```

2023-12-07

ADDED

Listing 41: squared

```
interface RequestData {  
    checksum?: string | boolean | ChecksumOutput; // Write base directory hashes  
    checksum?: 1; // Ignore recursive nested checksums  
}
```

Listing 42: squared.base

```
interface FileAsset {  
    checksum?: ChecksumValue; // Download  
    checksumOutput?: ChecksumValue; // Locally transformed  
}  
  
class Node  
    get floatClear(): ClearDirectionAttr;  
}
```

Listing 43: squared.lib.color

```
function intoRelativeRGB(method: string, relative: ColorRGB, r: string, g: string, b:
↳string, profile?: string): RGB | null; // Custom third-party override
```

Listing 44: android

```
interface DocumentOutput {
    dependencyScopes?: boolean | ArrayOf<"compile" | "provided" | "runtime" | "test" |
↳"snapshot">; // true is all except "snapshot"
}
```

Listing 45: android.lib.constant

```
const enum DEPENDENCY_TYPE {
    IMPLEMENTATION,
    API,
    COMPILE_ONLY,
    RUNTIME_ONLY,
    TEST_IMPLEMENTATION,
    ANDROID_TEST_IMPLEMENTATION
}
```

Listing 46: chrome

```
interface DocumentOutput {
    useSessionCache?: boolean;
}
```

MODIFIED

Listing 47: android.base

```
class Application {
    addDependency(group: string, name: string, type?: number, overwrite?: boolean): ↳
    ↳string; // type
    addDependency(group: string, name: string, version?: number, type?: number, ↳
    ↳overwrite?: boolean): string;
    addDependencyByProject(projectId: string, group: string, name: string, type?: number,
    ↳ overwrite?: boolean): string;
    addDependencyByProject(projectId: string, group: string, name: string, version?: ↳
    ↳string, type?: number, overwrite?: boolean): string;
}
```

2023-08-09

7.3.2 ADDED

Listing 48: squared.base

```
interface ElementSettings {
    excludeQuery?: ({ selector: string } & ExcludeOptions)[];
}

interface WatchInterval {
    start?: number | string;
}
```

Listing 49: android.base

```
interface UserResourceSettingsUI {
    convertLineHeight?: "px" | "dp" | "sp" | "pt" | "in" | "mm";
}
```

Listing 50: chrome.base

```
interface OutputCommand {
    rewrite?: boolean | URLData;
}
```

7.3.3 MODIFIED

Listing 51: squared.base

```
interface WatchInterval {
    interval?: number | string; // string
}

class NodeUI {
    getBoxSpacing(region?: BOX_STANDARD): [number, number, number, number]; // region
}
```


Listing 52: chrome

```
interface FileActionAttribute {
  preserveCrossOrigin?: boolean | URLData; // URLData
}
```

7.4 5.0.0

2023-04-29

7.4.1 BREAKING

Listing 53: squared.lib

```
/* util */

function splitEnclosing(value: string, pattern?: string | RegExp, start: number, opening?
↳: string, closing?: string): string[];

function splitEnclosing(value: string, pattern?: string | RegExp, count: number, opening?
↳: string, closing?: string): string[];
function splitEnclosing(value: string, pattern?: string | RegExp, options:
↳SplitEnclosingOptions, opening?: string, closing?: string): string[]; // start in
↳options

/* error */

function outputMessage(display: null, message: unknown, hint?: string): string;
function outputMessage(display: boolean, message: unknown, hint?: string): void;

function outputMessage(message: unknown, hint?: string, display?: boolean): void; //
↳display
```

Listing 54: squared.lib.error

```
function reject(value: string): Promise<never>;

function rejectError(value: string): Promise<never>;
```

Listing 55: squared.base

```
class Resource {
  addImage(resourceId: number, uri: string, options?: AddImageOptions): void; //
↳options
  getFonts(resourceId: number, fontFamily: string, options?: AddImageOptions):
↳FontFaceData[];
}

class ApplicationUI {
```

(continues on next page)

(continued from previous page)

```

    saveDocument(filename: string, content: string, options?: SaveDocumentOptions
↳<AppProjectUI>): void; // options
}

class ControllerUI {
    writeDocument(templates: NodeTemplate<NodeUI>[], depth: number, options?:
↳WriteDocumentOptions): string; // options
}

```

Listing 56: android.base

```

class Resource {
    static addTheme(resourceId: number, theme: ThemeAttribute, options?:
↳AddThemeOptions): boolean; // options
    static addString(resourceId: number, value: string, name?: string, options?:
↳AddStringOptions): string;
    static addImage(resourceId: number, images: StringMap, options?: AddImageOptions):
↳string;

    parseImage(resourceId: number, src: string, options?: ParseImageOptions): string; //
↳options
    addImageSrc(resourceId: number, element: HTMLImageElement | HTMLInputElement |
↳string, options?: AddImageSrcOptions): string;
}

```

7.4.2 ADDED

Listing 57: squared.base

```

class ResourceUI {
    parseBackgroundImage(node: NodeUI, value: string, outImage?: ImageSet[]):
↳ResourceBackgroundImage[] | undefined;
}

class Node {
    setParent(parent: Node | null, depth?: number, index?: number): void; // internalSelf
}

class NodeUI {
    get preIndent(): [string, NodeUI] | null;
    get causesLineBreak(): boolean;
}

```

Listing 58: squared.lib

```

/* color */

function intoRGB(value: ColorSpace): RGB | null; // External hook for CSS Color Module
↳Level 4

```

(continues on next page)

(continued from previous page)

```

/* constant */

const enum PLATFORM {
    CROS = 8,
    ANDROID = 16,
    IOS = 32
}

const enum USER_AGENT {
    CHROME_CROS = 16,
    CHROME_ANDROID = 64,
    SAFARI_IOS = 128
}

```

7.4.3 MODIFIED

Listing 59: squared.base

```

class Controller {
    get application(): Application<Node>;
}

class ControllerUI {
    get application(): ApplicationUI<NodeUI>;
}

class Resource {
    get application(): Application<Node>;
}

class ResourceUI {
    parseBackgroundImage(node: NodeUI, value: string, outImage?: ImageSet[]):
↳ResourceBackgroundImage[] | undefined;
    get application(): ApplicationUI<NodeUI>;
}

class ExtensionManager {
    get application(): Application<Node>;
}

class File {
    get resource(): Resource<Node>;
}

```


REFERENCES

8.1 squared

<https://unpkg.com/squared/types/type.d.ts>

ArrayOf
IteratorPredicate
PlainObject
StringMap

<https://unpkg.com/squared/types/object.d.ts>

Dimension

<https://unpkg.com/squared/types/dom.d.ts>

BoxMarginAttr
BoxModelAttr
BoxPaddingAttr
BoxRect
BoxRectDimension
ColorRGB
ContainerDimension
CssStyleAttr
CssStyleMap
DimensionAttr
FlexBox
FlexData
PseudoElt
PseudoStyleElt
PositionAttr

<https://unpkg.com/squared/types/squared.d.ts>

ExtensionRequest
ExtensionRequestObject
FileActionConfig
FileActionOptions
FileBroadcastOptions
FileObserveOptions
IncrementalData

PrefetchItem

<https://unpkg.com/squared/types/base/squared.d.ts>

APP_FRAMEWORK

APP_SECTION

BOX_STANDARD

NODE_PROCEDURE

NODE_RESOURCE

AppFramework

Application

ElementSettings

Extension

FileActionResult

MutationCallback

Node

NodeParentAttr

RootElement

<https://unpkg.com/squared/types/base/application.d.ts>

ControllerSettingsDirectoryUI

NodeTemplate

UserSettings

UserResourceSettings

UserResourceSettingsUI

<https://unpkg.com/squared/types/base/resource.d.ts>

URLData

<https://unpkg.com/squared/types/base/node.d.ts>

AutoMargin

CacheState

CacheStateUI

CacheValue

CacheValueUI

InitialData

LinearData

SupportUI

VisibleStyle

<https://unpkg.com/squared/types/base/file.d.ts>

BroadcastMessageCallback

BroadcastSocket

ChecksumOutput

DataSource

Exclusions

FileAsset

LogOptions

OutgoingHeaders

RequestData

ResponseData
TaskCommand
WatchInterval
WebSocketClient

<https://unpkg.com/squared/types/base/options.d.ts>

AlignedVerticallyOptions
AscendParameterOptions
BoxOptions
CoordsXYOptions
CssAnyOptions
CssAscendOptions
CssInitialOptions
CssSortOptions
CssUnitOptions
DescendParameterOptions
ExcludeOptions
HasOptions
HasUnitOptions
HideOptions
MinMaxOptions
NodeConvertUnitOptions
NodeParseUnitOptions
NodeUnitOptions
OffsetXYOptions
RemoveTryOptions
ReplaceTryOptions
SiblingsParameterOptions
TextHeightOptions
TranslateOptions
TraverseSiblingsOptions

<https://unpkg.com/squared/types/base/type.d.ts>

AnchorPositionAttr
BoxType
ClearDirectionAttr
DimensionSizableAttr
FloatDirectionAttr
OrientationAttr
ReferenceType

<https://unpkg.com/squared/types/lib/squared.d.ts>

ListIterator

<https://unpkg.com/squared/types/lib/data.d.ts>

ElementData
UserAgentData

<https://unpkg.com/squared/types/lib/options.d.ts>

ContainerCascadeOptions
ContainerFindOptions
ContainerIncludesOptions
ContainerRangeOptions
ContainerRemoveIfOptions

<https://unpkg.com/squared/types/lib/type.d.ts>

Specificity

<https://unpkg.com/squared/types/svg/object.d.ts>

SvgTransformExclude

<https://unpkg.com/squared/types/internal/squared.d.ts>

ObserveSocket

8.2 Android

<https://unpkg.com/squared/types/android/squared.d.ts>

ATTRIBUTE_COLOR
BUILD_VERSION
DEPENDENCY_TYPE
RESOURCE
SYSTEM_COLOR
AppViewModel

<https://unpkg.com/squared/types/android/application.d.ts>

ControllerSettingsDirectoryUI
ControllerSettingsUI
ThemeAttribute
UserResourceSettingsUI

<https://unpkg.com/squared/types/android/resource.d.ts>

FontProviderFonts
ManifestData
ResourceQualifierMap

<https://unpkg.com/squared/types/android/file.d.ts>

DocumentOutput
RequestData

<https://unpkg.com/squared/types/android/extension.d.ts>

ExtensionAttributeElement
ExtensionFragmentElement
ExtensionIncludeElement
ExtensionSubstituteElement
ExtensionViewModelElement

<https://unpkg.com/squared/types/android/node.d.ts>

Constraint

LocalSettingsUI

NamingStyles

<https://unpkg.com/squared/types/android/data.d.ts>

ViewAttribute

<https://unpkg.com/squared/types/android/options.d.ts>

AddArrayOptions

AddImageOptions

AddStringOptions

AddThemeOptions

IsAnchoredOptions

CloneOptions

<https://unpkg.com/squared/types/android/type.d.ts>

DependencyScopes

LayoutChainStyle

LayoutDirectionAttr

LayoutGravityAttr

LayoutGravityDirectionAttr

ResolutionUnit

8.3 Chrome

<https://unpkg.com/squared/types/chrome/application.d.ts>

UserResourceSettings

<https://unpkg.com/squared/types/chrome/file.d.ts>

DocumentOutput

RequestData

TemplateMap

<https://unpkg.com/squared/types/chrome/data.d.ts>

UnusedAtRule

UsedStylesData

<https://unpkg.com/squared/types/chrome/options.d.ts>

SaveAsOptions

<https://unpkg.com/squared/types/chrome/type.d.ts>

ResourceHintType

8.4 E-mc

<https://unpkg.com/@e-mc/types/lib/core.d.ts>

PermittedDirectories

8.5 TypeScript

<https://github.com/microsoft/TypeScript/blob/main/src/lib/es5.d.ts>

InstanceType

<https://github.com/microsoft/TypeScript/blob/main/src/lib/dom.generated.d.ts>

CSSStyleDeclaration

DOMRect

DOMStringMap

Element

HTMLElement

MutationObserverInit

ShadowRoot

<https://github.com/microsoft/TypeScript/blob/main/src/lib/es2015.iterable.d.ts>

Iterable

IterableIterator

8.6 MDN

<https://developer.mozilla.org/docs/Web/API/Request/mode>

RequestMode

VERSIONS

| Date | Docs | squared | E-mc | sqd-cli |
|----------|-------|---------|-------|---------|
| 05-03-24 | 0.1.1 | 5.2.1 | 0.9.1 | 0.14.1 |
| 04-29-24 | 0.1.0 | 5.2.0 | 0.9.0 | 0.14.0 |

LICENSE

Apache 2.0

A

add() *(built-in function)*, 29
 addDependency() *(built-in function)*, 44
 addDependencyByProject() *(built-in function)*, 44
 addFontProvider() *(built-in function)*, 45
 addXmlNs() *(built-in function)*, 45
 android *(module)*, 43
 appendFiles() *(built-in function)*, 38
 appendTo() *(built-in function)*, 35
 apply() *(built-in function)*, 31
 attr() *(built-in function)*, 31
 auth() *(built-in function)*, 27

B

broadcast() *(built-in function)*, 27

C

chrome *(module)*, 49
 clear() *(built-in function)*, 29
 close() *(built-in function)*, 34
 copyFiles() *(built-in function)*, 38
 copyTo() *(built-in function)*, 36
 customize() *(built-in function)*, 46

E

extend() *(built-in function)*, 28

F

findDocumentNode() *(built-in function)*, 33
 fromElement() *(built-in function)*, 41
 fromNode() *(built-in function)*, 41

G

get() *(built-in function)*, 30
 getElementById() *(built-in function)*, 39
 getLocalSettings() *(built-in function)*, 48

K

kill() *(built-in function)*, 27

L

latest() *(built-in function)*, 34
 loadCustomizations() *(built-in function)*, 46

O

observe() *(built-in function)*, 41
 observeSrc() *(built-in function)*, 42

P

parseDocument() *(built-in function)*, 32
 parseDocumentSync() *(built-in function)*, 33
 prefetch() *(built-in function)*, 32

Q

querySelector() *(built-in function)*, 40
 querySelectorAll() *(built-in function)*, 40

R

remove() *(built-in function)*, 30
 removeObserver() *(built-in function)*, 48, 49
 reset() *(built-in function)*, 34
 resetCustomizations() *(built-in function)*, 47

S

save() *(built-in function)*, 34
 saveAs() *(built-in function)*, 35
 saveCustomizations() *(built-in function)*, 47
 saveFiles() *(built-in function)*, 37
 setEndpoint() *(built-in function)*, 26
 setFramework() *(built-in function)*, 28
 setHostname() *(built-in function)*, 26
 setLocalAddress() *(built-in function)*, 26
 setResolutionByDeviceName() *(built-in function)*, 47
 setViewModel() *(built-in function)*, 43
 setViewModelByProject() *(built-in function)*, 43