

CDS JavaScript Client library Developer's Guide

Status: [draft]

EasyRP team <easyrp-eng@google.com>

To help developers to interact with CDS server that based on CDS Specification, we provide this client library. It hides most of CDS technical details, like JSON-RPC packing/unpacking, cross domain messaging, and provides a very simple API for a site to interact with CDS.

1. Introduction

1.1 Initialize a CdsClient

To call a service on the CDS, you need a CdsClient instance. Then, before you call a CDS service, you must initialize the CdsClient.

You can do both by below code:

```
var client = window.accountchooser.CdsClient.init({
  popupMode: false,
  cdsServer: {           // Optional CDS server config.
    domain: "",           // CDS domain. e.g, 'https://www.accountchooser.biz'
    iframe: "",           // Path for iframe page. e.g., '/iframe.html'
    popup: "",            // Path for popup page. e.g., '/popup.html'
    redirect: ""          // Path for redirect page. e.g., '/redirect.html'
  },
  clientCallbackUrl: 'http://gitkit-rp.appspot.com/cds-tester.htm',
  callbacks: {
    store: done
  }
});
```

Or, if you like, you can write separate code for each task as below:

```
var client = new window.accountchooser.CdsClient ({
  ...
});
client.init();
```

1.2 Start a service

Next you can call CDS services directly. Below code will start a UX flow for the store service.

```
client.store(account, localAccounts);
```

Below code will start a UX flow for the select service.

```
client.select(localAccounts);
```

Below code will start a UX flow for the update service.

```
client.update(account);
```

By calling the service, the user will be redirected to a CDS page to complete it. And decided by the *popupMode* in the config, a popup window or full page redirect can be used.

1.3 Return the result

After making choice in the CDS page, the user will be redirected back to the site page again. If popup mode is used, the popup window could be closed automatically or just kept, controlled by the *config.keepPopup*. The dark screen in the site page will be removed. If full page redirect is used, the page will be redirected back.

In both case, the callback method in the *config.callbacks.{service}* will be invoked. For example, in below example, *done* method will invoked after store service returned.

```
var client = window.accountchooser.CdsClient.init({  
  popupMode: false,  
  ... ..  
  callbacks: {  
    store: done  
  }  
});
```

Note: In full page redirect mode, after the page returned to the site, the callback method will be invoked by the client.init() method. In other words, the client.init() method not only initialize the CdsClient according to the config, but also check whether there is any returned response. If there is any response, the corresponding callback of that response will be invoked; if no response, *empty* callback will be invoked instead. See below example.

```
var client = window.accountchooser.CdsClient.init({  
  popupMode: false,  
  ... ..  
  callbacks: {  
    store: storeCallback,  
    select: selectCallback,  
    update: updateCallback,  
    empty: emptyCallback  
  }  
});
```

2. Configuration parameters

2.1 popupMode

Should popup window be used when starting a CDS service?

Values:

- *true*: use popup window.
- *false*: use full page redirect.

Optional: yes. Default value is *false*.

2.2 popupWindow

A popup window object to be used when starting a CDS service. Only used in popup mode.

Values: A popup window object.

Optional: yes.

2.3 cdsServer

CDS server location.

Value: An object with multiple properties.

Optional: yes. If omit, use default value of each properties.

Sample:

```
cdsServer: {           // Optional CDS server config.  
  domain: "",          // CDS domain.  
  iframe: "",          // Path for iframe page. e.g., '/iframe.htm'  
  popup: "",           // Path for popup page. e.g., '/popup.htm'  
  redirect: ""         // Path for redirect page. e.g., '/redirect.htm'  
},
```

2.3.1 cdsServer.domain

The CDS Server domain.

Value: A Domain Name.

Optional: yes. Default value is: <https://www.accountchooser.com>

2.3.2 cdsServer.iframe

Path for CDS iframe page.

Values: A relative URL, starting with '/'.

Optional: yes. Default value is [/iframe.html](#).

2.3.3 cdsServer.redirect

Path for CDS popup page.

Values: A relative URL, starting with '/'.

Optional: yes. Default value is [/redirect.html](#).

2.3.4 cdsServer.popup

The CDS URL that a site should open in popup mode.

Values: A relative URL, starting with '/'.

Optional: yes. Default value is [/popup.html](#).

2.4 clientCallbackUrl

The site URL to which CDS returns result.

Values: A URL.

Optional: yes. If omit, current site URL (includes the query parameters part) will be used.

2.5 callbacks

The handlers for services.

Values: an Object with key-value pairs. The key is a CDS service name, while the value is a callback function to handle the return value of that service.

Optional: yes.

2.6 ui templating ([see overview](#))

CDS customized UI configuration.

Value: an object with multiple properties. A sample is show below.

Optional: yes. If omit, no customized UI shown.

Sample:

```
ui: {  
  favicon: "",           // URL of the favicon.  
  title: "",             // The window title.  
  header: "",            // URL of the header HTML.  
  main: {  
    url: "",              // URL of the main HTML.  
    method: "             // Methods for the main HTML: sanitize | cajole.  
                          // Note: cajole is only supported on accountchooser.biz now;  
  
    enableOnMobile:      // Forces the main part show on mobile device: true | false.  
  },  
  footer: "",            // URL of the footer HTML.  
}
```

2.6.1 ui.favicon

The URL for the favorite icon.

Values: a URL.

Optional: yes. If omit, the CDS favorite icon will be shown.

2.6.2 ui.title

The title of the window.

Values: A string.

Optional: yes. If omit, the CDS page title ‘Central Account Chooser’ will be shown.

2.6.3 ui.header

The URL for the header HTML.

Values: a URL.

Optional: yes. If omit, no header will be shown.

2.6.4 ui.footer

The URL for the footer HTML.

Values: a URL.

Optional: yes. If omit, no footer will be shown.

2.6.5 ui.main.url

The URL for the main HTML (marketing text).

Values: a URL.

Optional: yes. If omit, no marketing text will be shown.

2.6.6 ui.main.method

Methods for the main HTML: sanitize | cajole.

Values: a predefined method name string.

Optional: yes. Default value is ‘sanitize’. [Note: cajole is only supported on accountchooser.biz now]

2.6.7 ui.main.enableOnMobile

Forces the main part show on mobile device.

Values: boolean.

Optional: yes. Default value is ‘false’.

2.7 popupWidth

The width of the popup window. Only used in popup mode.

Values: an integer.

Optional: yes. Default value is 520.

2.8 popupHeight

The height of the popup window. Only used in popup mode.

Values: an integer.

Optional: yes. Default value is 550.

2.10 keepPopup

Whether to close the popup window automatically or not after the service is done. Only used in popup mode.

Values:

- *true*: keep the popup window.
- *false*: automatically close the popup window.

Optional: yes. Default value is *false*.

2.11 showAll

Whether to show accounts with an email field only or all accounts in CDS. [**Open issue**: this parameter should be renamed to a more precise name.]

Values:

- *true*: show all accounts.
- *false*: show accounts with an email field only.

Optional: yes. Default value is *false*.

2.12 idpFilter

A list of IDP domains. CDS will only show accounts belonging to these IDP domains when this value is set to a non-empty list.

Values: an Array of IDP domains.

Optional: yes.

2.13 language

The language used in CDS UI. [not supported yet.]

Values: en, fr, zh_CN, ...

Optional: yes. Default value is en.

3. methods summary

Method	Description
<i>CdsClient(config)</i>	Constructor. Create an client instance.
<i>init()</i>	Initialize the CdsClient.
<i>static init(config)</i>	Create and initialize the CdsClient.
<i>store(account, localAccounts, opt_cdsOptions)</i>	Start CDS store service.
<i>select(localAccounts, opt_cdsOptions)</i>	Start CDS select service.
<i>update(account, opt_cdsOptions)</i>	Start CDS update service.
<i>changePopupModeTo(mode)</i>	Change CdsClient's popup mode after it initialized
<i>setPopupWindow(popupWindow)</i>	Set the popup window used in popup mode.
<i>getPopupWindow()</i>	Get the popup window used in popup mode.
<i>closePopupWindow()</i>	Close the popup window.

4. Initialize CdsClient

4.1 Constructor: CdsClient(config)

Description: create a CdsClient object with the configuration.

```
var client = new window.accountchooser.CdsClient (config);
```

4.2 instance method: CdsClient.init()

Description: initialize the CdsClient object with the configuration. In full page redirect mode the returned service result will be checked, and invoked if any.

```
client.init();
```

4.3 static method: CdsClient.init(config)

Description: constructs a CdsClient object, and initialize it. It is a wrapper method for above 2 methods.

```
var client = window.accountchooser.CdsClient.init (config);
```

5. CDS services

5.1 store(account, localAccounts, opt_cdsOptions)

Description: start CDS 'store' service to let user confirm whether he wants to store 'account' into the CDS.

Parameters:

- **account:** The account to be stored. It has 1 required property 'email', and 3 optional parameters: 'displayName', 'photoUrl' and 'providerId'.
- **localAccounts:** The accounts saved in local site. CDS can show them in UI so that user can select them. It is an array of object, each object has 1 required property 'email', and 3 optional parameters: 'displayName', 'photoUrl' and 'providerId'.
- **opt_cdsOptions:** Options for CDS. These options will be merged with global ones (set by the initializing config) and passed to CDS. These are one-time and won't change global options. This parameter is optional. Options supported are listed as below:

```
{
  clientCallbackUrl: '',    // Address returned from CDS.
  keepPopup: true/false,  // Whether to keep the popup window or not.
}
```

```
client.store(account, []);
```

5.2 select(localAccounts, opt_cdsOptions)

Description: start CDS 'select' service to let user select 1 account (among local accounts and CDS accounts) to return.

Parameters:

- **localAccounts:** The accounts saved in local site. CDS can show them in UI so that user can select them. It is an array of object, each object has 1 required property 'email', and 3 optional parameters: 'displayName', 'photoUrl' and 'providerId'.
- **opt_cdsOptions:** Options for CDS. These options will be merged with global ones (set by the initializing config) and passed to CDS. These are one-time and won't change global options. This parameter is optional. Options supported are listed as below:

```
{
  clientCallbackUrl: '',    // Address returned from CDS.
  keepPopup: true/false,  // Whether to keep the popup window or not.
  showAll: true/false,    // Whether to show all accounts.
  idpFilter: [],          // A list of IDP domains.
}
```

-

```
client.select([account1, account2]);
```

5.2 update(account, opt_cdsOptions)

Description: start CDS 'update' service to let user confirm whether he wants to update 'account' in the CDS.

Parameters:

- **account:** The account to be updated. It has 1 required property 'email', and 3

- optional parameters: 'displayName', 'photoUrl' and 'providerId'.
- opt_cdsOptions: Options for CDS. These options will be merged with global ones (set by the initializing config) and passed to CDS. These are one-time and won't change global options. This parameter is optional. Options supported are listed as below:

```
{
  clientCallbackUrl: '', // Address returned from CDS.
  keepPopup: true/false, // Whether to keep the popup window or not.
}
```

```
client.update(account);
```

6. CDS return values

After returned from CDS, you callback will be invoked with two parameters:

- result: the result of the service. It should be a JavaScript object.
- error: the error object (defined by json-rpc), if some error occurs.

Result and error shouldn't be exist at the same time.

6.1 result: {store: true/false}

Description: Only effective to CDS 'store' service. If user confirms to store the 'account' into CDS, *{store: true}* will be returned. If user cancel the store request, *{store: false}* will be returned.

6.2 result: {account: {email: "", displayName: "", photoUrl:"}}

Description: If user click an account in the account list, this result will be returned.

6.3 result: {account: {email: "", displayName: "", photoUrl:"}, idpAssertion: {}}

Description: If user click an account in the account list and this account is belong to an IDP which CDS can interact with invisibly, this result will be returned. The idpAssertion is an object returned from IDP, including account email, provider ID and so on. **[Note: cajole is only supported on accountchooser.biz now, also the format of idpAssertion is undefined.]**

6.4 result: {addAccount: true}

Description: If user click 'sign in to another account' button, this result will be returned.

6.5 result: {updated: true/false}

Description: Only effective to CDS 'update' service. If user confirms to update the 'account' in CDS, *{updated: true}* will be returned. Otherwise, *{store: false}* will be returned.

6.6 error: {code: -1, message: "", detail:"}

Description: If some error occurs, this result will be returned.

7. Other methods

7.1 changePopupModeTo(popupMode)

Description: allows the CdsClient changes to the new popup mode after it initialized.

Parameters:

- popupMode: The new popup mode.

```
client.changePopupModeTo(true);
```

7.2 setPopupWindow(popupWindow)

Description: sets the popup window used in popup mode. If the user doesn't set the popup window, CdsClient will open a popup automatically and the user can get it by `getPopupWindow()`.

Parameters:

- popupWindow: The popup window object.

```
client.setPopupWindow(popup);
```

7.3 getPopupWindow()

Description: gets the popup window used in popup mode.

Returns: the popup window used in popup mode.

```
var popup = client.getPopupWindow();
```

7.4 closePopupWindow()

Description: If the popup window exists, then close it. Otherwise do nothing.

```
client.closePopupWindow();
```

8. Language selector

A site can show a language selector in the (footer or header part of its) customized UI on the CDS page by following below steps.

1) Add a <select> with id attribute set to 'AC_LANGUAGE_LIST'.

2) For each <option> in it, set a URL as its value:

`<option value="http://a-url?a=b" >English</option>`

the page will be redirected to the URL if user select this option. The URL will be validated by below rules:

- Only HTTP/HTTPS schema allowed.
- Must be an absolutely URL (starting with: https:// or http)
- No hash allowed.

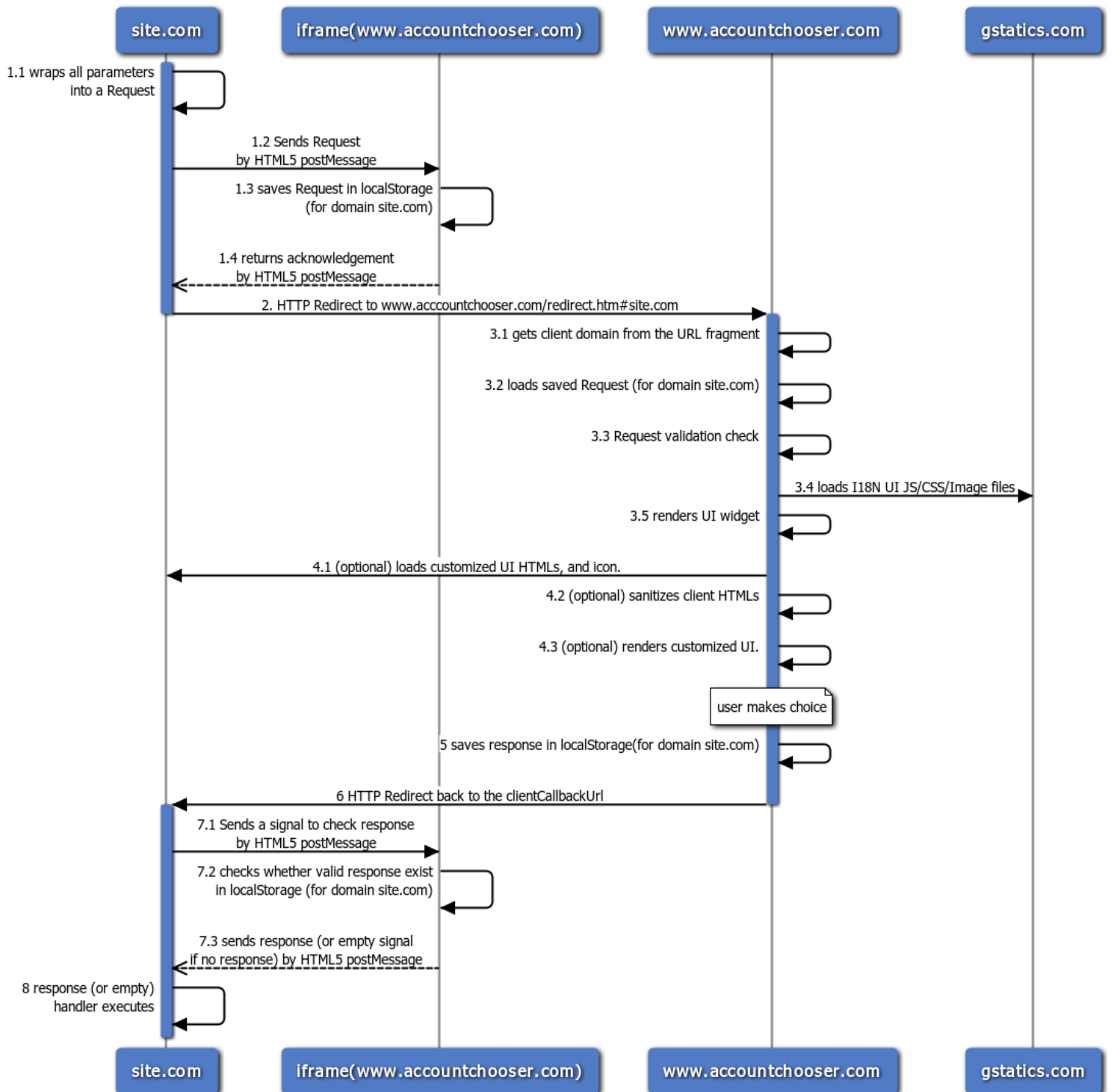
See an example on <https://ac-test.appspot.com/footer.jsp>

3) Normally, a handler on the target URL should resubmit a request to CDS with new language parameter and new customized UI URLs (so that both the CDS account list and the customized UI are changed to the new language).

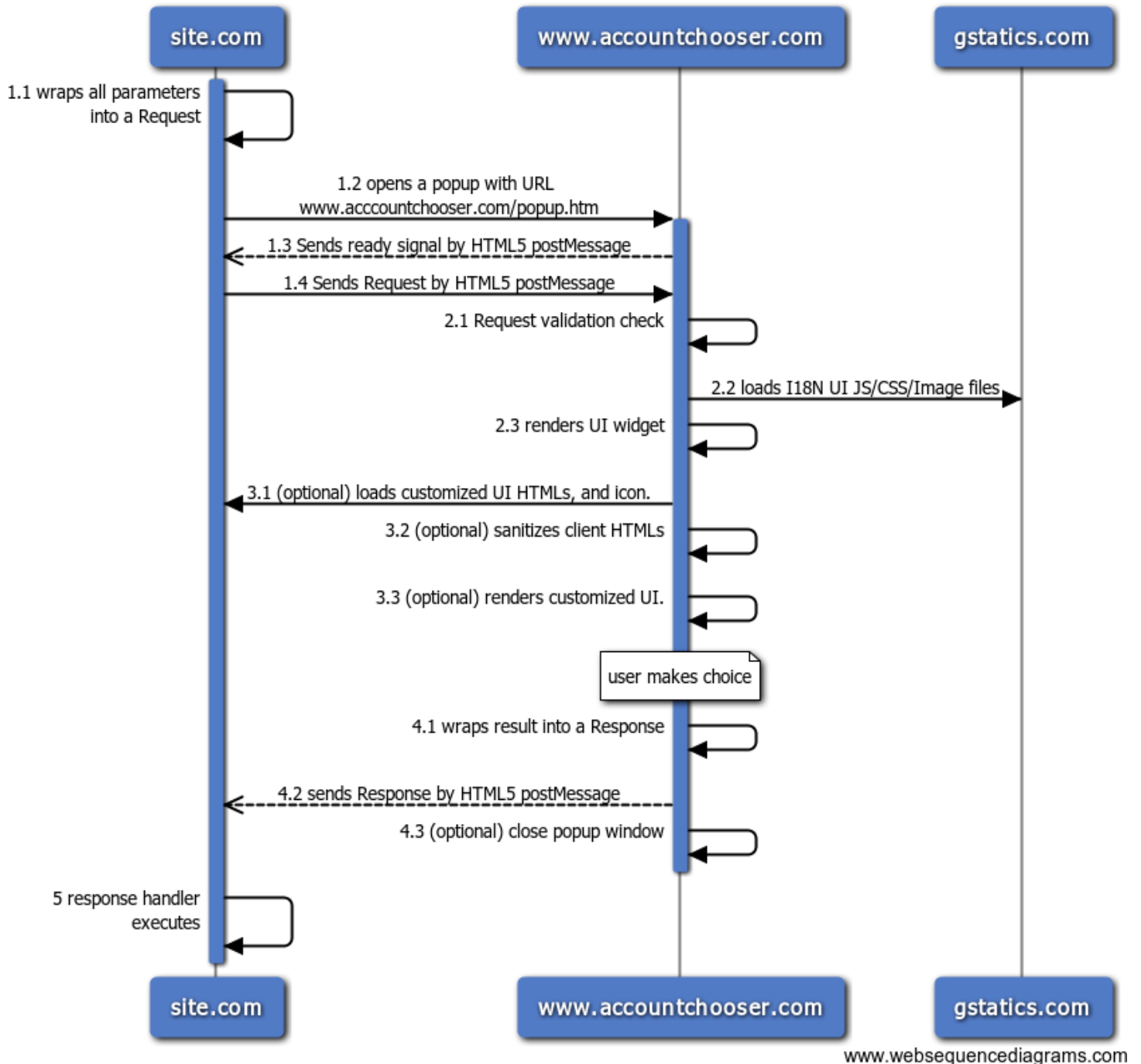
9. Sequence diagrams

Below diagrams shows the overall steps how a site interacts with the CDS.

Redirect Mode sequence diagram



Popup Mode sequence diagram



One note for the popup mode 3.1, the popup window loads these resource from the server side of site.com (instead of the main window), so no `postMessage` needed.

10. References

- [1] [In-browser CDS Specification - Draft](#)
- [2] [Demo site using full page redirect mode.](#)
- [3] [Demo site using popup mode.](#)
- [4] [Demo site for mobile browser.](#)