

# Vue.js - 1.2

## On this page:

- [Extension ID](#)
- [What's new?](#)
- [Description](#)
- [In what situation should you install this extension?](#)
- [Supported Vue.js versions](#)
- [Supported template definitions](#)
- [CAST AIP compatibility](#)
- [Prerequisites](#)
- [Dependencies with other extensions](#)
- [Download and installation instructions](#)
- [What results can you expect?](#)
  - [Objects](#)
  - [Store](#)
    - [Dispatch](#)
  - [Webservices](#)
    - [Axios](#)
    - [Vue-resources](#)
    - [Fetch](#)
  - [Link transition](#)
  - [Quality rules](#)
  - [Known limitations](#)
    - [Store map function](#)
    - [Functions passed as properties](#)

## Target audience:

Users of the extension providing **Vue.js** support for Web applications.



**Summary:** This document provides basic information about the extension providing **Vue.js** support for Web applications.

## Extension ID

`com.castsoftware.vuejs`

## What's new?

Please see [Vue.js - 1.2 - Release Notes](#) for more information.

## Description

This extension provides support for **Vue.js**. Vue.js is an open-source JavaScript framework used to build user interface for web application: [vuejs.org](https://vuejs.org)

## In what situation should you install this extension?

If your Web application contains **Vue.js** source code and you want to view these object types and their links with other objects, then you should install this extension.

## Supported Vue.js versions

Version	Support	Comment
v2.x		

## Supported template definitions

Vue.js has different methods to define a component's template - these methods are shown here: <https://vuejsdevelopers.com/2017/03/24/vue-js-component-templates/>. The following table shows the methods that are currently supported by Vue.js extension:

Pattern	Support
String	✗
Template literal	✗
X-Templates	✓
Inline	✓
Render functions	✓
JSX	✗
Single page components	✓



Unsupported template definition methods will not create links to any called functions.

# CAST AIP compatibility

This extension is compatible with:

CAST AIP release	Supported
8.3.x	

## Prerequisites

- An installation of any compatible release of CAST AIP (see table above)

## Dependencies with other extensions

Some CAST extensions require the presence of other CAST extensions in order to function correctly. The **Vue.js** extension requires that the following other CAST extensions are also installed:

- [HTML5/JavaScript](#)
- **Web services linker service** (internal technical extension)

When using the **CAST Extension Downloader** to download the extension and the **Manage Extensions** interface in **CAST Server Manager** to install the extension, any dependent extensions are **automatically** downloaded and installed for you. You do not need to do anything.

## Download and installation instructions

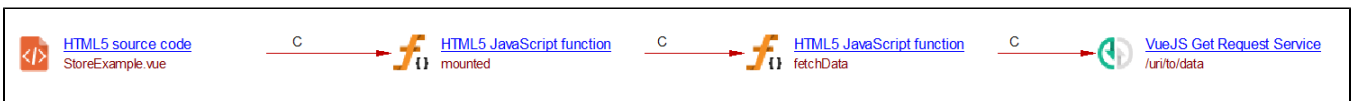
Please see:

- [Download an extension](#)
- [Install an extension](#)

The latest [release status](#) of this extension can be seen when downloading it from the CAST Extend server.

## What results can you expect?

Once the analysis/snapshot generation has completed, you can view the results in the normal manner (for example via CAST Enlighten):



## Objects

The following specific objects are displayed in CAST Enlighten:

Icon	Description
	Vue.js Get Request Service
	Vue.js Post Request Service
	Vue.js Put Request Service

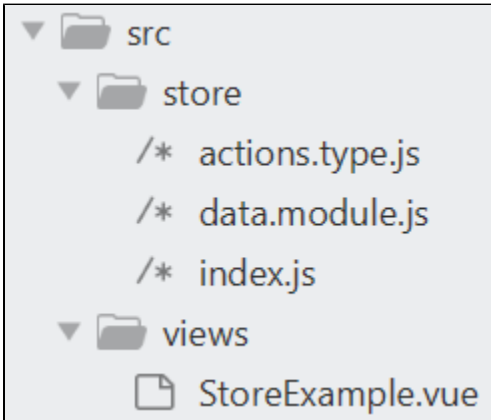


## Store

### Dispatch

Vue.js extension support Vuex store dispatch methods are used to call shared function.

Here is an example of a classical architecture using this method.



#### In actions.type.js

```
export const FETCH_DATA = "fetchData";
```

#### In data.module.js

```
import { FETCH_DATA } from "../actions.type";
import Vue from "vue";
import axios from "axios";
import VueAxios from "vue-axios";

Vue.use(VueAxios, axios);

const actions = {
  [FETCH_DATA]() {
    return return Vue.axios.get('uri/to/data')
  },
};

export default {
  actions
};
```

### In index.js

```
import Vue from "vue";
import Vuex from "vuex";

import data from "../data.module";

Vue.use(Vuex);

export default new Vuex.Store({
  modules: {
    data
  }
});
```

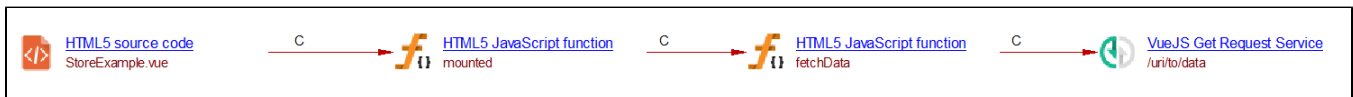
### In StoreExample.vue

```
<template>
  <div></div>
</template>

<script>
import { FETCH_DATA } from "@store/actions.type";

export default {
  name: "StoreExample",
  mounted: function() {
    this.$store.dispatch(FETCH_DATA)
  }
};
</script>
```

In Enlighten:



## Webservices

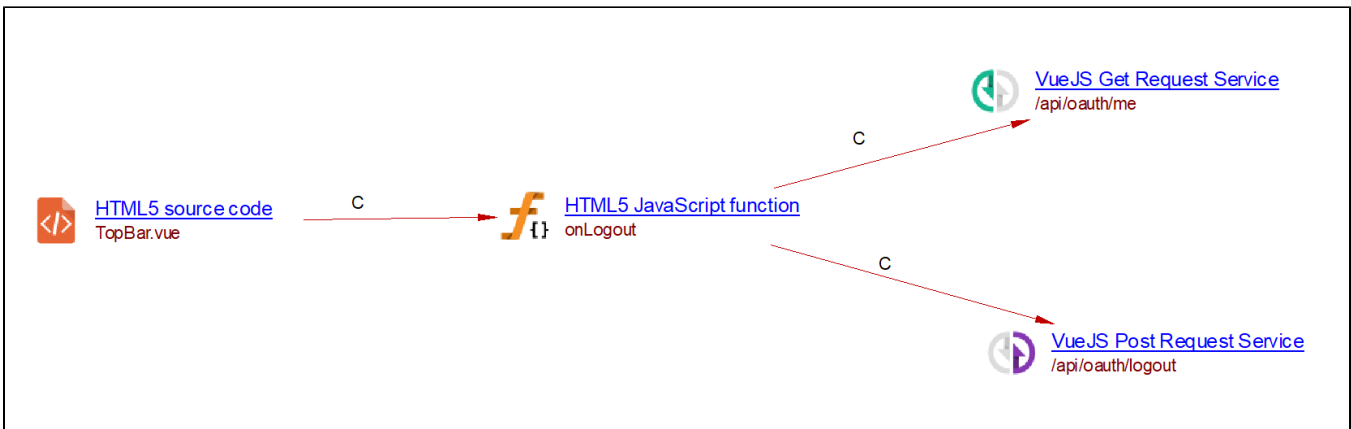
### Axios

The Vue.js extension supports webservises using Axios:

```
async onLogout() {
  await this.$axios.$post('/api/oauth/logout');
  this.setLogout(true);
  this.setCurrentUser(null);

  try {
    await this.$axios.$get('/api/oauth/me')
  } catch (e) {
    // middleware redirects to login page
  }
}
```

Giving the following graph in Enlighten:



## Vue-resources

The Vue.js extension supports webservices using vue-resources.

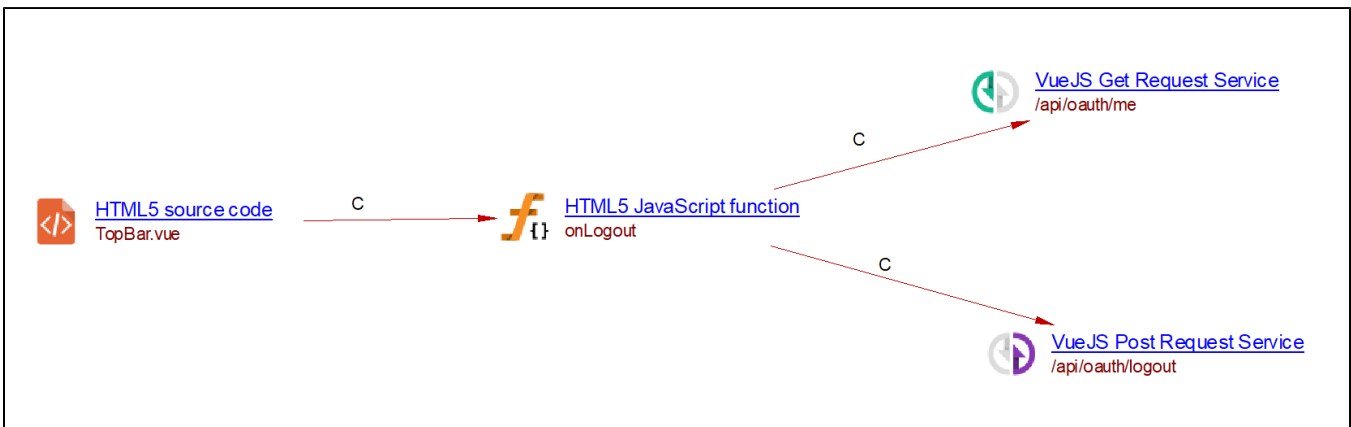
```

async onLogout() {
  await this.$http.post('/api/oauth/logout');
  this.setLogout(true);
  this.setCurrentUser(null);

  try {
    await this.$http.$get('/api/oauth/me')
  } catch (e) {
    // middleware redirects to login page
  }
}

```

Giving the following graph in Enlighten:



## Fetch

The Vue.js extension supports webservices using fetch:

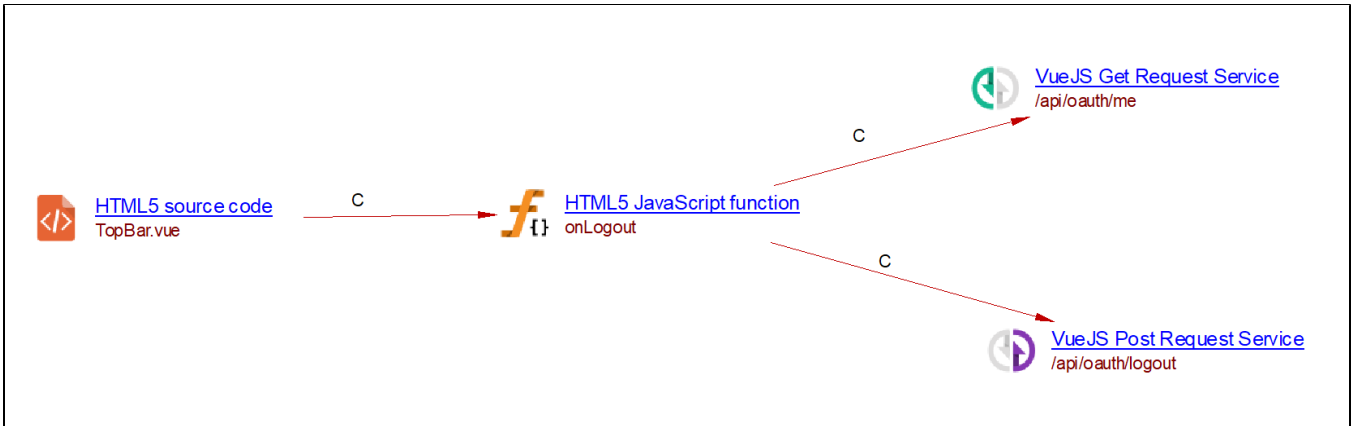
```

async onLogout() {
  await fetch('/api/oauth/logout');
  this.setLogout(true);
  this.setCurrentUser(null);

  try {
    await fetch('/api/oauth/me')
  } catch (e) {
    // middleware redirects to login page
  }
}
}

```

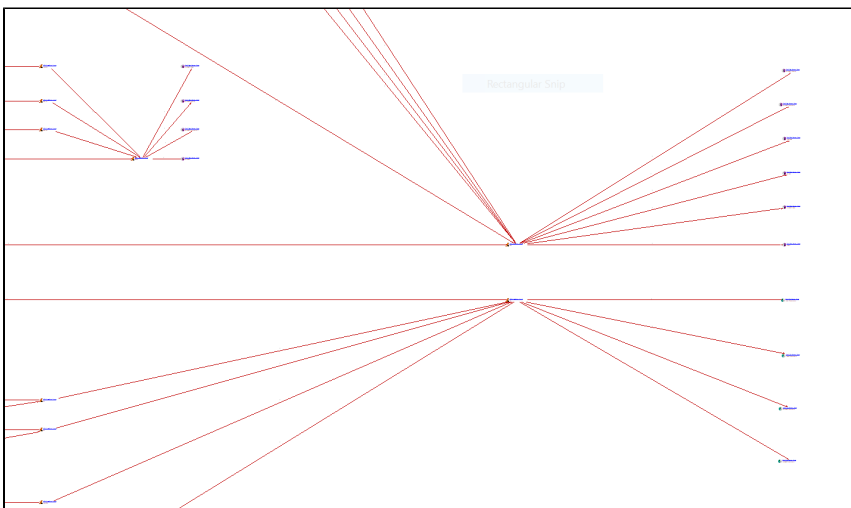
Giving the following graph in Enlighten:



## Link transition

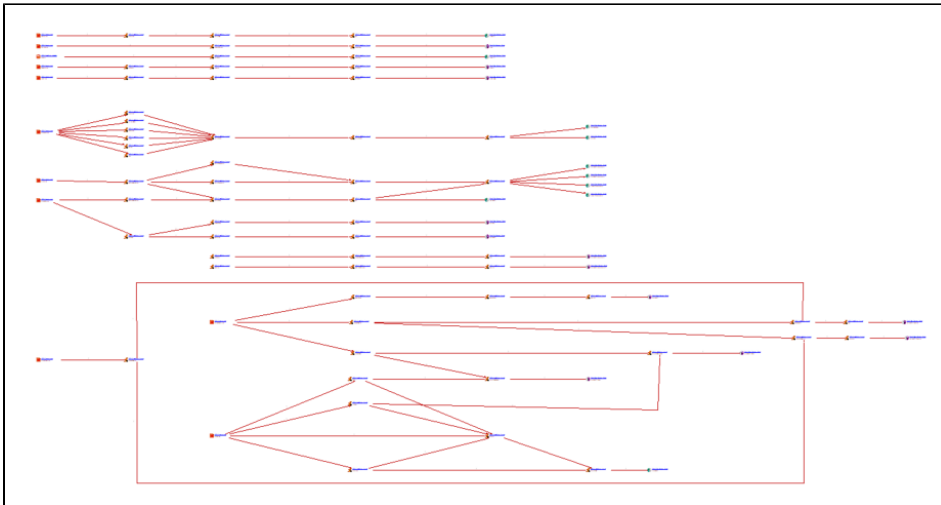
Sometimes calls to services are factorized into a function that only concatenates URI parts leading to the following kinds of bottleneck and making it hard to follow which function called a given service:

*Click to enlarge*



Link transition allows the extension to go through the specified functions to make a direct link between the function defining the service URI and the service object. This leads to a better graph without bottlenecks:

*Click to enlarge*



## Quality rules

The following structural rules are provided:

<b>1.2.1-funcrel</b>	<a href="https://technologies.castsoftware.com/rules?sec=srs_vuejs&amp;ref=  1.2.1-funcrel">https://technologies.castsoftware.com/rules?sec=srs_vuejs&amp;ref=  1.2.1-funcrel</a>
<b>1.2.0-funcrel</b>	<a href="https://technologies.castsoftware.com/rules?sec=srs_vuejs&amp;ref=  1.2.0-funcrel">https://technologies.castsoftware.com/rules?sec=srs_vuejs&amp;ref=  1.2.0-funcrel</a>

## Known limitations

### Store map function

Currently there are no links to the following store map functions from components using them:

- mapGetters
- mapActions

For example:

#### In SettingsModal.vue

```
// ...
<ToggleSwitch v-model="user.autosave_notes" @change="setAutosaveNotes(user.autosave_notes)" />
// ...
methods: {
  ...mapActions(['deleteUser', 'setShowLanguageTags', 'setAutosaveNotes', 'revokeUserAccess']),
  // ...
}
```

#### In user.js

```
const actions = {
  // ...
  setAutosaveNotes({ commit }, flag) {
    return client.put('/user/autosave-notes', { flag }).then(({ data }) => {
      commit(SET_USER, data)
    })
  }
}
```

No link shown in CAST Enlighten:





[HTML5 source code](#)  
SettingsModal.vue



[HTML5 JavaScript function](#)  
setAutosaveNotes

## Functions passed as properties

In Vue.js components can have properties, which could be functions. Links to those functions are currently unsupported.