AngularJS - 1.0

On this page:

Target audience:

Users of the extension providing **AngularJS** support for Web applications.

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

Description

This extension provides support for **AngularJS**.

In what situation should you install this extension?

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

- Creates AngularJS object structure (see below)
- Creates links between these objects and links between HTML files and controller scope elements
- Directives are partly handled:
  - Links between HTML tags and directives are handled
  - Links between HTML attributes in directive sections and directive scope items are handled

Supported AngularJS versions

The following table displays the supported versions matrix:

Version	Transaction Support	Rules Support	Type of Support
1.0	✅	✅	Fullstack use case support
1.2	✅	✅	Fullstack use case support
1.3	✅	✅	Fullstack use case support
1.4	✅	✅	Fullstack use case support
1.5	✅	✅	Fullstack use case support

Function Point, Quality and Sizing support

This extension provides the following support:

- **Function Points (transactions)**: a green tick indicates that OMG Function Point counting and Transaction Risk Index are supported
- **Quality and Sizing**: a green tick indicates that CAST can measure size and that a minimum set of Quality Rules exist

Function Points (transactions)	Quality and Sizing
✅	✅

Comparison with existing support for JavaScript in CAST AIP

CAST AIP has provided support for analyzing JavaScript via its **J2EE and .NET analyzers** (provided out of box in CAST AIP) for some time now. The **HTML5/JavaScript extension** (on which the **AngularJS** extension depends) also provides support for JavaScript but with a focus on web applications. CAST highly recommends that you use this extension if your Application contains JavaScript and more specifically if you want to analyze a web application, however you should take note of the following:

- You should ensure that you configure the extension to NOT analyze the back end web client part of a .NET or J2EE application.
- You should ensure that you configure the extension to ONLY analyze the front end web application built with the HTML5/JavaScript that communicates with the back end web client part of a .NET or J2EE application.
- If the back end web client part of a .NET or J2EE application is analyzed with the AngularJS extension and with the native .NET/J2EE analyzers, then your results will reflect this - there will be duplicate objects and links (i.e. from the analyzer and from the extension) therefore impacting results and creating erroneous Function Point data.

CAST AIP compatibility

This extension is compatible with:

CAST AIP release	Supported
8.2.x	✅
8.1.x	✅
8.0.x	✅
7.3.4 and all higher 7.3.x releases	✅

Supported DBMS servers

This extension is compatible with the following DBMS servers:

CAST AIP release	CSS2	Oracle	Microsoft
All supported releases	✅	✅	❌

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 **AngularJS** extension requires that the following other CAST extensions are also installed:

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

> ⓘ Note that 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:

- http://doc.castsoftware.com/display/EXTEND/Download+an+extension
- http://doc.castsoftware.com/display/EXTEND/Install+an+extension

> ⓘ The latest release status of this extension can be seen when downloading it from the CAST Extend server.

Packaging, delivering and analyzing your source code

Please see: AngularJS - Packaging, delivering and analyzing your source code

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):
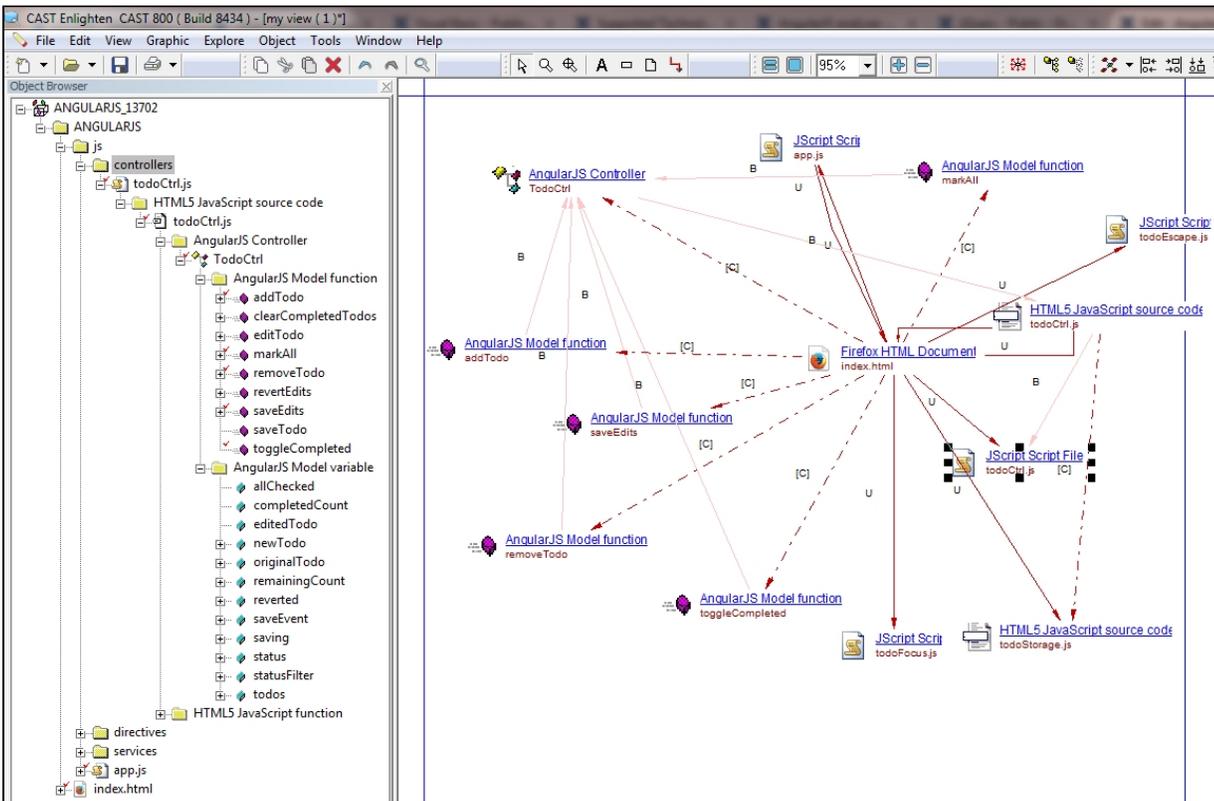
Objects

The following objects are displayed in CAST Enlighten:

Icon	Description
	JavaScript file
	HTML file (icon depends on default browser)
	HTML5 JavaScript function object
	HTML5 JavaScript Source Code object
	AngularJS Controller
	AngularJS Directive
	AngularJS Directive Controller
	AngularJS Directive Link Function
	AngularJS Factory
	AngularJS Model function
	AngularJS Model variable
	AngularJS Get Resource Service
	AngularJS Post Resource Service
	AngularJS Put Resource Service

🐢	AngularJS Delete Resource Service

Rules

You can find a full list of rules delivered with this extension here: https://technologies.castsoftware.com/?rlH=extensions/com.castsoftware.angularjs/1.0.0-funcrel.json

Technical Notes

AngularJS Service objects and Transactions

AngularJS Service objects are not considered by CAST AIP as being part of any transaction (they are considered as technical function points). Instead, the **AngularJS Service's functions** are considered to be part of the transaction. Therefore when looking at the results of an AngularJS analysis, links to the AngularJS Service's functions should be checked to ensure that the expected transactions exist. Transactions within AngularJS are ALWAYS from functions or methods to functions or methods (objects which are executable).

Known Limitations

In this section we list the most significant functional limitations that may affect the analysis of applications using AngularJS:

- The extension does not detect AngularJS objects when they are declared with prototype. For example:

```
xxx.service('ContentService', ['$rootScope', '$window', '$resource', '$q', '$log', 'ManageContentURIs',
attContent.content]);

attContent.content.prototype.retrieveContent = function(criteria, useStubs) {
var useStub = useStubs?useStubs:false;
var deferred = this.q.defer(),
```