Dgeni Doc-Gen Tool Improvements

*Status: (Draft)*

*Authors: pete@bacondarwin.com*

*This document is published to the web as part of the public* [Angular Public](https://drive.google.com/open?id=0BxgtL8yFJbacQmpCc1NMV3d5dnM&authuser=0)/[Design Docs](https://drive.google.com/open?id=0BxgtL8yFJbacUnUxc3l5aTZrbVk&authuser=0)/[Document Generation](https://drive.google.com/open?id=0B1zR3solO5-wflFkRjBZeGE2YTRtUE9qTXBDVHNWNUhHR0lHc3RwY3NPWHRVdC1ZbjZGZUU&authuser=0) *folder*

# Objective

This document suggests some design improvements to the dgeni and dgeni-packages projects to support the forthcoming Angular 2 project and related sub-projects, router, i18n and ngAnimate.

# Background

Dgeni and its associated dgeni-packages project has been effective as a maintainable documentation generation tool for the Angular 1 project. But it has some configuration deficiencies when maintaining a larger project and also it would benefit from improved support for generating docs across multiple related projects, such as would be the case for Angular 1 + 2 and the router.

# Prior Art

* dgeni
* dgeni-packages
* jsdoc

# Detailed Design

## Document Types

Currently documents in dgeni are just plain JS objects, with a property called docType, which is used to distinguish one kind of document from another. This makes them very flexible but also makes it difficult to reason about what properties and methods one can expect a document to possess.

Moreover, a number of properties of a document are computed from other properties. Currently this is done in various Processors but the parameters that change the way these Processors affect different types of document are in config blocks of different Packages. This means it is not always clear what the final Processor configuration will be for a particular type of document.

An example of this is computing the id and path properties for a document.

Going forward each document could be an instance of a specific class, that can provide methods for computing properties that are specific to the document.

Bear in mind that processors would still need to be responsible for adding and removing documents and high level modifications to the document collection.

## Generic API Document Specification

To allow documents to be used in host container applications it would be good if we could create a specification for an API document that can be shared across different host applications.