

Mapping Model Identifiers to Storage Locations

Joop Ringelberg

27-08-21, 20-07-22

Version: 2

Introduction

The storage service provider is free to choose its own scheme for mapping URLs in a models subdomain to a Couchdb database, as long as it obeys two restrictions:

- URLs with a path that starts with "models" should be mapped to a store that stores just models;
- URLs with a path that starts with "cw" should be mapped to a store that stores just instances.

An author needs to maintain her model and this involves creating versions that are not accessible to the public. Yet, with the mapping from model identifier to storage location, we seem to have precluded this practice.

We restore this possibility with an extra restriction on the mapping of model identifiers to storage locations, to be carried out by the providers of a storage service.

Pre-release information in semantic version numbers

Semantic versioning provides for [pre-release information](#):

A pre-release version MAY be denoted by appending a hyphen and a series of dot separated identifiers immediately following the patch version. Identifiers MUST comprise only ASCII alphanumerics and hyphens [0-9A-Za-z-]. Identifiers MUST NOT be empty. Numeric identifiers MUST NOT include leading zeroes. Pre-release versions have a lower precedence than the associated normal version. A pre-release version indicates that the version is unstable and might not satisfy the intended compatibility requirements as denoted by its associated normal version. Examples: 1.0.0-alpha, 1.0.0-alpha.1, 1.0.0-0.3.7, 1.0.0-x.7.z.92, 1.0.0-x-y-z.-.

Including pre-release information in the mapping

We map a model identifier to a URL using a simple scheme. The identifiers under the model scheme are:

```
Identifier = "//" reg-name "/" segment-nz-nc "@" <version core> [ -<br><pre-release>]
```

We map them to URLs using the following rule:

```
model://{host}/{modelname}@{semver}<br>=><br>https://f("models", prerelease({semver})).{host}/{modelname}@{semver}
```

where the function `prerelease` maps the semantic version string to its first alphanumeric string and `f` is a function from the string "models" and the result of `prerelease`.

What exactly the function `f` is, is to the discretion of the storage service provider. For example:

```
https://models.perspect.it/System@1.1.0-alpha.1
```

may be mapped to:

```
/alpha_models/System@1.1.0-alpha.1
```

or, just as good, to:

```
/models_alpha/System@1.1.0-alpha.1
```

In effect, this gives the modeller control over numerous repositories to be used for pre-release versions.

In other words: from the pre-release string we take the first part (the part up to the first "."). This we use to construct a database name: it is the database that holds the pre-release versions of models.

The storage provider may want to limit its clients in the number of databases they can create. It can require its clients to create these databases before using them (databases will not be created on-the-fly). As a consequence, the modeller must be conscientious in choosing and using `<pre-release>`.

Also, note that for these databases to be private, a modeller must use a pre-release name that is unique within the Couchdb.