

Tags Subdomain

# Table of Contents

Screenshots .....	2
Update tag .....	3
New tags .....	5
How to configure/use .....	7
Classpath .....	7
Bootstrapping .....	7
API .....	8
Known issues .....	9
Dependencies .....	10

This module (`isis-module-tags`) provides the ability to add multiple tags (or labels) to any domain entity.

For example, an `Email` entity could be associated with a `priority` tag of 'urgent', 'normal' or 'low', and could also have a `category` tag of 'work', 'family', 'friends'.

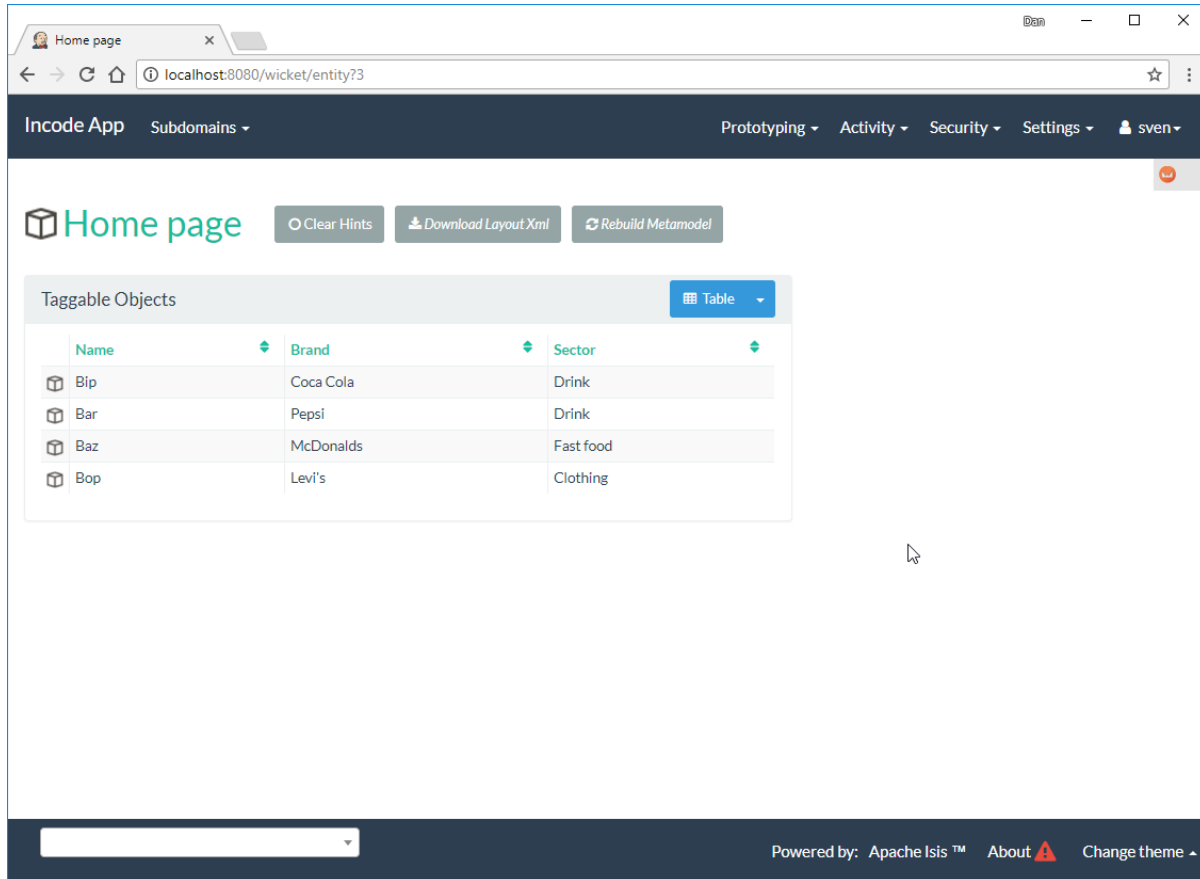
The idea is to allow a user to label an entity by offering a choice of other existing tags associated with other entities of the same type and also tag type. That is, the combination of the entity type (eg `com.mycompany.myapp.Email`) and the tag type (eg `priority`) creates a named set of available tags (eg 'urgent', 'normal', 'low').

As well as listing existing tags, new tags can of course also be created, and existing tags updated or removed.

# Screenshots

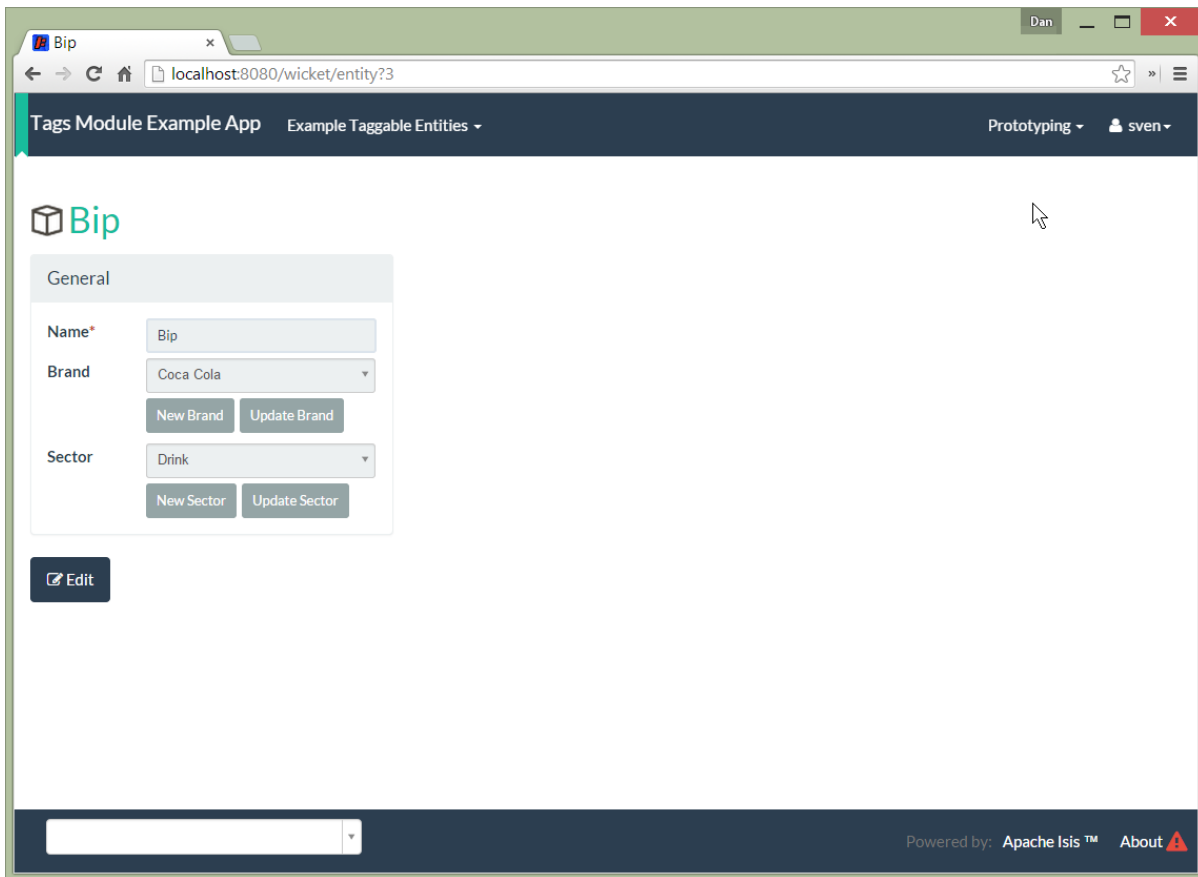
The module's functionality can be explored by running the [quickstart with example usage](#) using the `org.incode.domainapp.example.app.modules.ExampleDomDomTagAppManifest`.

A home page is displayed when the app is run:



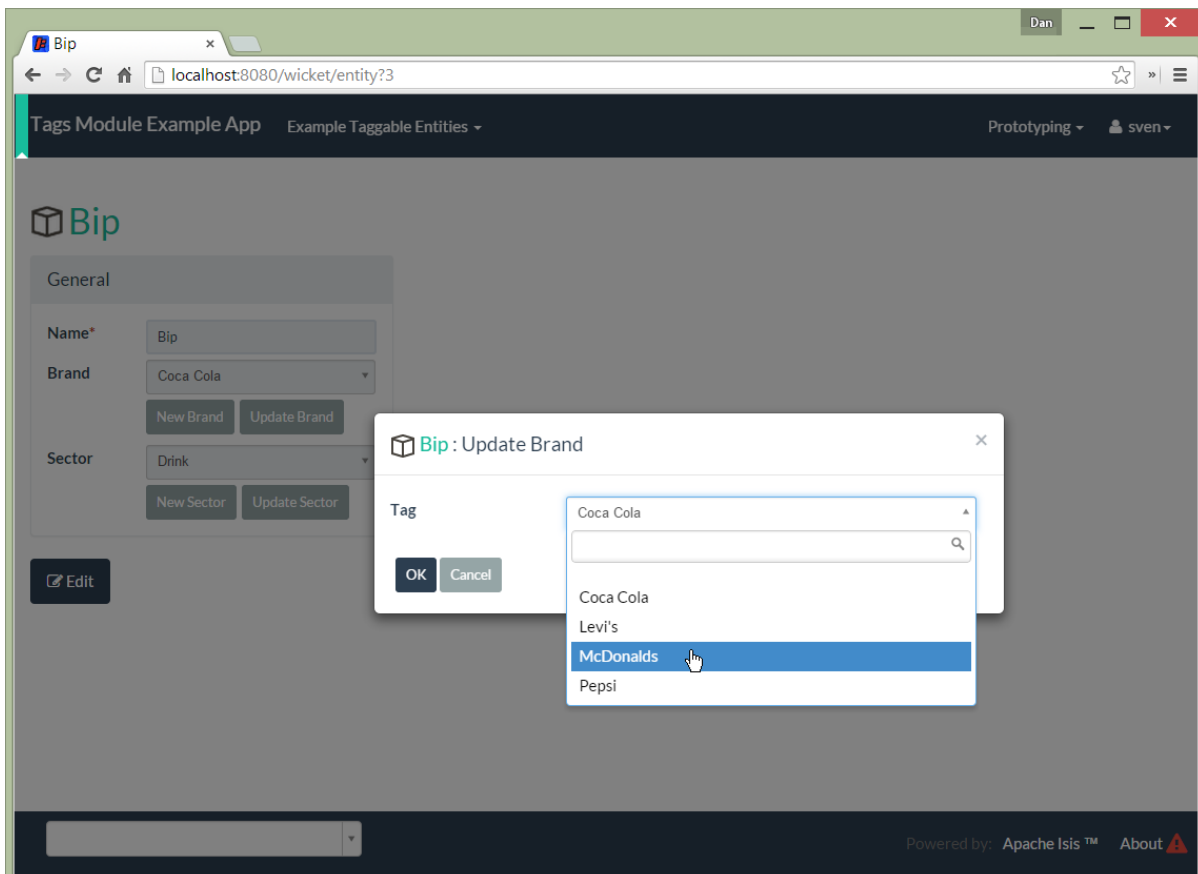
The remaining screenshots below **do** demonstrate the functionality of this module, but are out of date in that they are taken from the original `isisaddons/incodehq` module (prior to being amalgamated into the `incode-platform`).

One of the demo objects created is "Bip". The class of this entity is not of interest, what *is* of interest is that it has two properties, **brand** and **sector**, that correspond to ``Tag`s`:

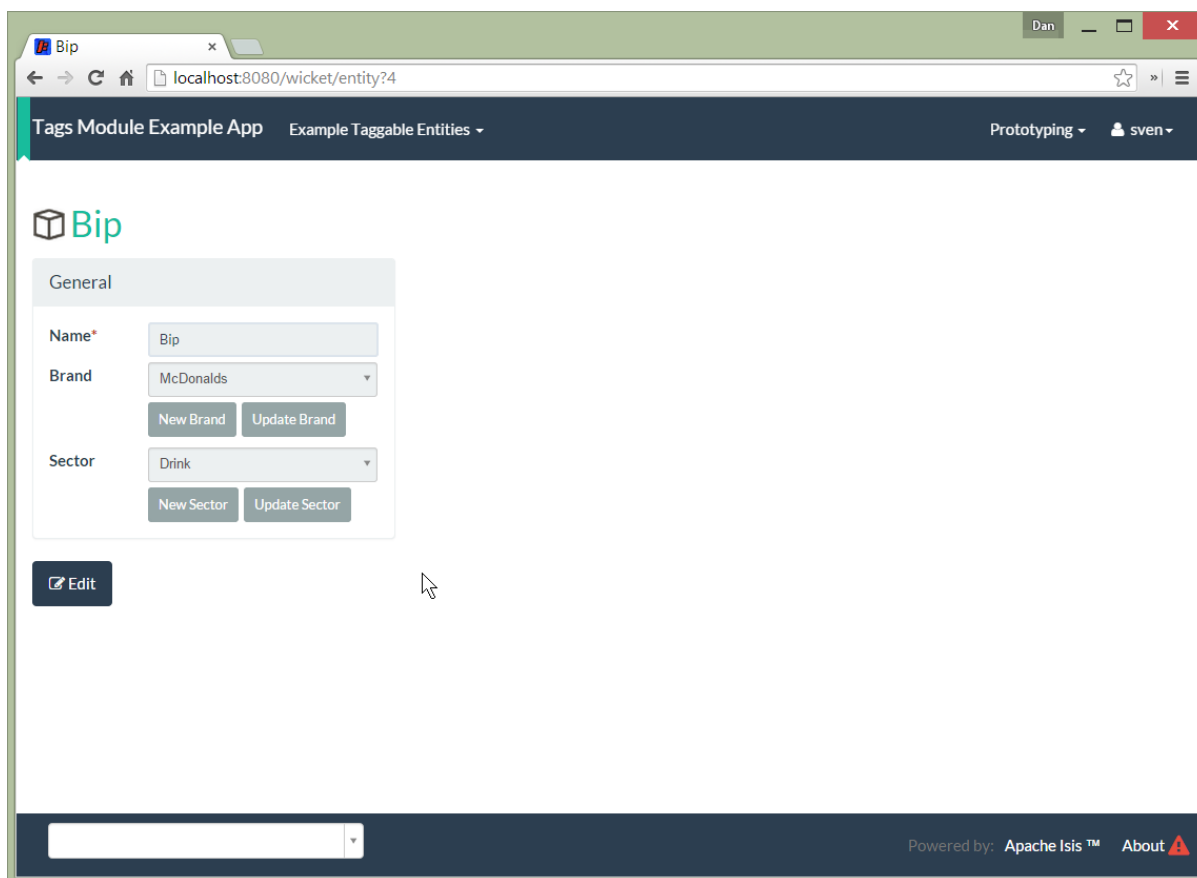


## Update tag

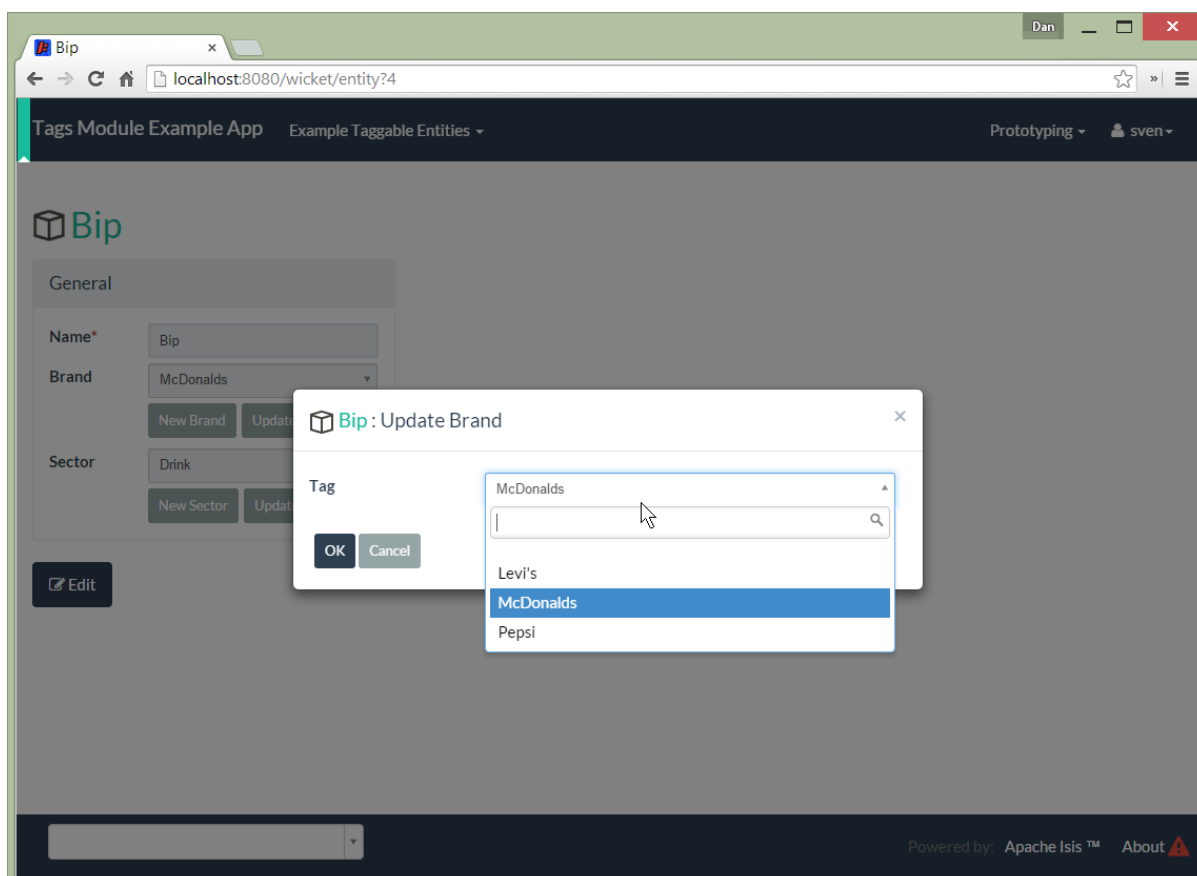
Using the action, can update tag. The **Tags** service provides a list of available tag (names) for this entity type and tag name ("brand"):



The tag can then be saved:

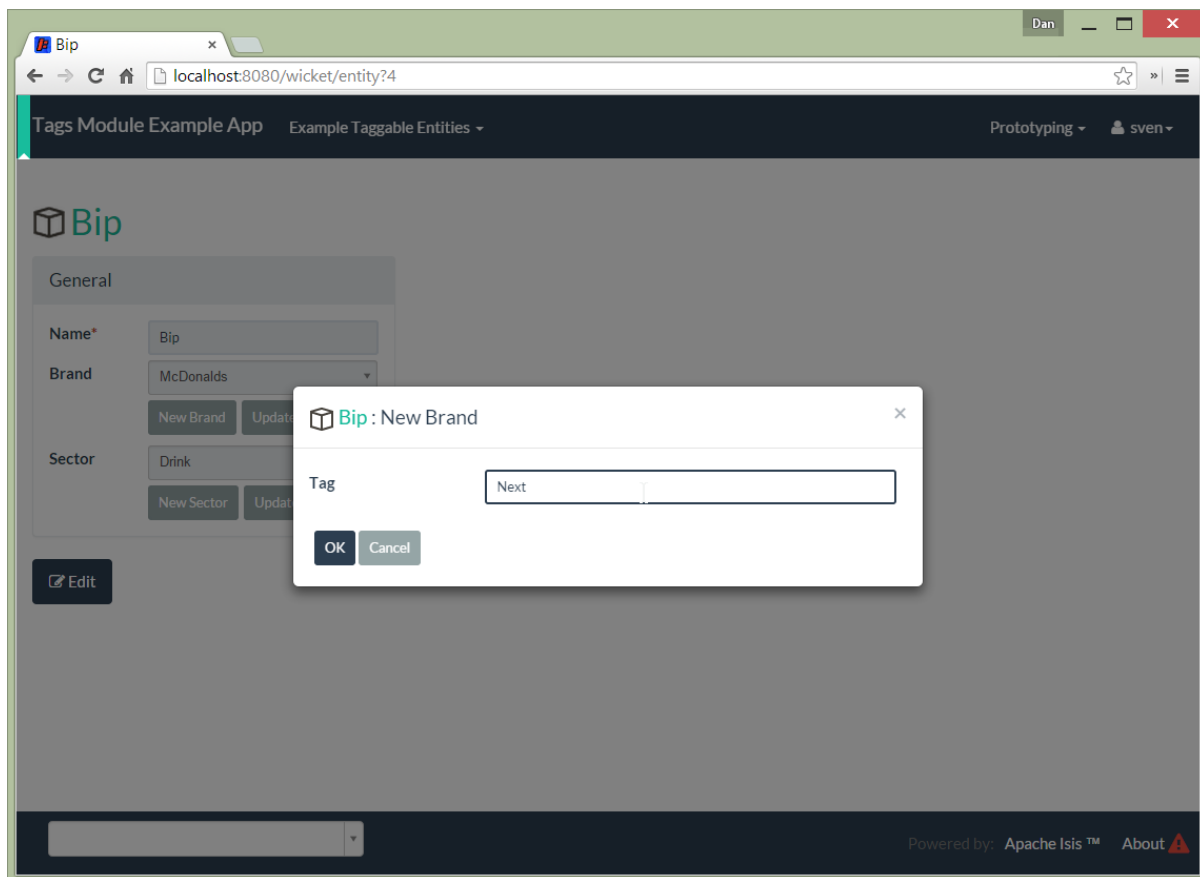


If the tag is updated once more, note that the original tag ("Coca Cola") is no longer listed. This is because there are no other entities that have this label:

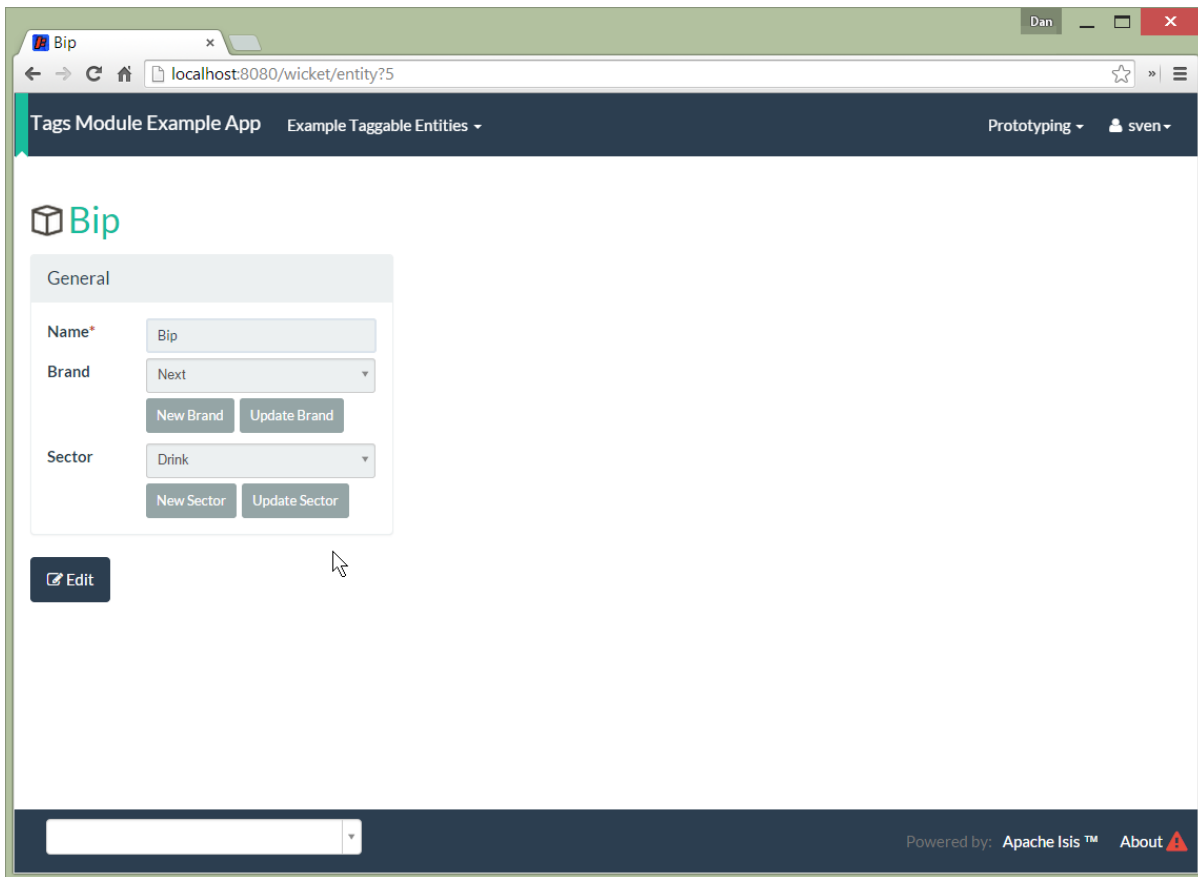


# New tags

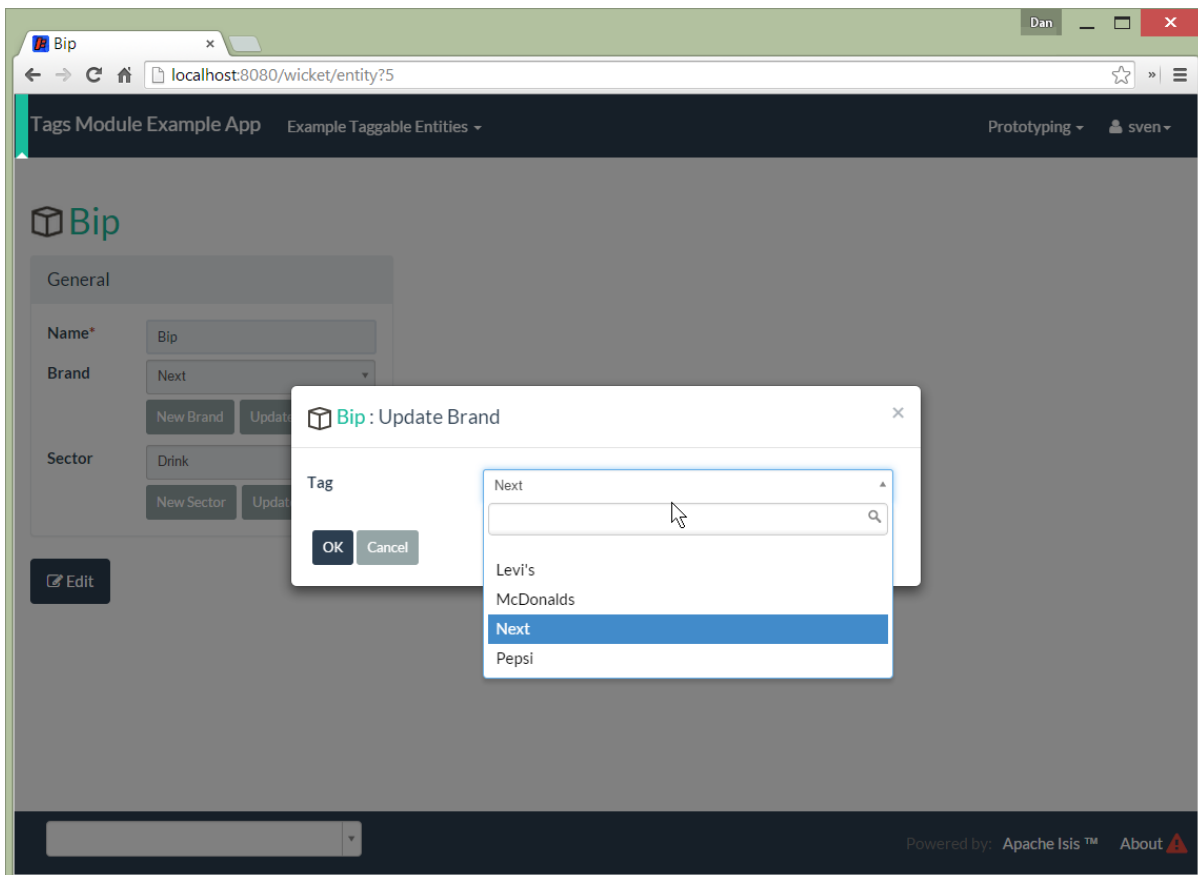
New tags can also be created. Enter the tag name:



The tag is saved:



... and is available as a choice for this or any other entity of the same type:





# How to configure/use

## Classpath

Update your classpath by adding this dependency in your dom project's `pom.xml`:

```
<dependency>
  <groupId>org.isisaddons.module.tags</groupId>
  <artifactId>isis-module-tags-dom</artifactId>
  <version>1.15.0</version>
</dependency>
```

Check for later releases by searching [Maven Central Repo](<http://search.maven.org/#search|ga|1|isis-module-tags-dom>).

For instructions on how to use the latest **-SNAPSHOT**, see the [contributors guide](#).

## Bootstrapping

In the `AppManifest`, update its `getModules()` method, eg:

```
@Override
public List<Class<?>> getModules() {
    return Arrays.asList(
        ...
        org.isisaddons.module.tags.TagsModule.class,
        ...
    );
}
```

# API

The main functionality is exposed through the **Tags** domain service, which provides two methods:

```
public class Tags {  
  
    @Programmatic  
    public List<String> choices(  
        final Object taggedObject, final String tagKey) { ... } ①  
  
    @Programmatic  
    public Tag tagFor(  
        final Object taggedObject,  
        final Tag existingTag,  
        final String tagKey, final String tagValue) { ... } ②  
  
}
```

<1> `choices(...)` returns a list of tags as strings for the object being tagged (eg **Email**) and for the tag key (eg **priority**) <2> **tagFor(...)** will create/update a tag for the object being tagged (eg **Email**), the tag key (eg **priority**) and the tag value (eg 'urgent'). The existing tag (if any) is passed in so that it can be removed if the tag value is left as null.

# Known issues

None known at this time.

# Dependencies

Maven can report modules dependencies using:

```
mvn dependency:list -o -pl modules/dom/tags/impl -D excludeTransitive=true
```

which, excluding the Apache Isis modules, returns no direct compile/runtime dependencies.

The module *does* use icons from [icons8](#).