LearnCubiByExamplesC1
Learn Cubi By Examples Chapter 1 - Application Design
id , **en**

Updated Oct 23, 2011 by rockys...@gmail.com

Learn Cubi By Examples Chapter 1 - Application Design

This book uses Cubi Trac to demonstrate the best practice of building Cubi applications. First we need to understand the business requirements of the Trac application.

Business requirements

Trac is a ticketing system that can be used in various business environment.

- For product development, Trac can be the bug tracking system.
- For customer service, Trac can be used to track the customer requests.
- Trac may be also used as a component of bigger applications such as project management, office automation, help desk and so on.

Features

The main function of Trac is to manage tickets. A ticket should have the following attributes

- Id
- Name or Title
- Summary
- Description or Content
- Type (Defect | Feature | Task | ...)
- Status (Open | Accepted | Reopened | Resolved | Closed)
- Priority (P0 | P1 | P2 | P3 |...)
- Severity (High | Medium | Low)
- Product
- Product component
- Version
- Milestone
- Resolution (Unresolved | Fixed | Won't Fix | Duplicated | Incomplete | Cannot reproduce | ...)
- Ticket reporter
- Ticket owner
- Ticket creation time
- Ticket update time

A ticket can have

- attachments
- user comments
- change history

The application should allow users to

- create a new ticket
- edit ticket attributes
- add comments on a ticket
- view change history
- attach files to ticket
- search tickets and save the search query

Users

Three types of users can access Trac with different permissions.

Unregistered user may

- view a ticket, but not change or comment a ticket
- search ticket
- view comments of a ticket

Registered user may do all actions an unregistered user does and

- create a ticket

- add comment to a ticket
- edit a ticket
- store a ticket query
- attach files to a ticket

Administrator may do all actions a registered user does and

- manage the enumeration of ticket status
- manage the enumeration of ticket priority
- manage the enumeration of ticket severity
- manage the enumeration of ticket resolution
- manage products and components
- manage version
- manage milestone
- manage users

Pages design

Having the business requirements, it is time to draw the pages, also called mockups.

As Cubi already provides commonly used pages (user registration, my account, ...), let's focus on Trac specific pages.

Ticket pages

We will need the following page to manage tickets

- Ticket listing page (with paging, filters)
- Ticket creation page
- Ticket details page (show attachments and change history)
- Ticket search page (search by name, product, priority, ...)
- My Tickets page

Administration pages

The pages needed for administrating tickets include

- Product management page (product name and description)
- Product component management page (component name, description and owner)
- Version management page (version number and description)
- Milestone management page (milestone name, due time, complete time and description)
- Enumeration management page (customize priorities, types, severity, status and resolutions)

Menu and Tab

Now that we have defined the application pages, we will need to add links of these pages to the navigation system - tabs and menus.

Tabs are the top level links appear under page header. Menus are the second level links of the specific Tab. With Trac added into Cubi, the navigation will be like:

- Administration (Cubi Tab)
  - Application Menu
    - User
    - Role
    - ...
- Ticket (Trac Tab)
  - Browse Tickets
    - All Tickets
    - Search Tickets
    - Create Ticket
    - My Tickets
    - Ticket reported by me
  - Trac Admin
    - Products
    - Components
    - Versions
    - Milestones
    - Enumerations

Database design

Having feature requirements and page mockups, it is the time for designing the database. Let's have a prefix 'trac' for all tables of Trac application.

The most important table is trac_ticket of course.

```
CREATE TABLE `trac_ticket` (
  `id` int(11) NOT NULL auto_increment,
```

```
  `type` varchar(30) default NULL,
  `time` datetime default NULL,
  `changetime` datetime default NULL,
  `product_id` int(11) default NULL,
  `component_id` int(11) default NULL,
  `severity` varchar(30) default NULL,
  `priority` varchar(30) default NULL,
  `owner_id` int(11) default NULL,
  `reporter_id` int(11) default NULL,
  `cc` varchar(255) default NULL,
  `version_id` int(11) default NULL,
  `milestone_id` int(11) default NULL,
  `status` varchar(30) default NULL,
  `resolution` varchar(30) default NULL,
  `summary` varchar(255) default NULL,
  `description` text,
  `keywords` varchar(128) default NULL,
  PRIMARY KEY  (`id`),
  KEY `Index1` (`summary`),
  KEY `Index2` (`type`),
  KEY `Index3` (`product_id`),
  KEY `Index4` (`component_id`),
  KEY `Index5` (`priority`),
  KEY `Index6` (`owner_id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

We also need tables to store ticket's comments and changes. We want to store comment and changes in a serialized string in 'comments' column.

```
CREATE TABLE `trac_comments` (
  `id` int(11) NOT NULL auto_increment,
  `parent_id` int(11) NOT NULL,
  `create_time` datetime default NULL,
  `author_id` int(11) default NULL,
  `comments` text,
  PRIMARY KEY  (`id`),
  KEY `Index1` (`parent_id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

Table to store "saved searches"

```
CREATE TABLE `trac_query` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(128) NOT NULL,
  `search_rules` text NOT NULL,
  `owner_id` int(11) DEFAULT NULL,
  `public` int(2) DEFAULT '0',
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8
```

Other supporting tables include:

- trac_product
- trac_component
- trac_version
- trac_milestone
- trac_enumeration

```
CREATE TABLE `trac_product` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(128) NOT NULL DEFAULT '',
  `description` text,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;


CREATE TABLE `trac_component` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(128) NOT NULL DEFAULT '',
  `product_id` int(11) NOT NULL,
  `owner_id` int(11) DEFAULT NULL,
  `description` text,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;


CREATE TABLE `trac_version` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(30) NOT NULL DEFAULT '',
  `update_time` datetime DEFAULT NULL,
  `description` text,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;


CREATE TABLE `trac_milestone` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(128) NOT NULL DEFAULT '',
  `due` datetime DEFAULT NULL,
  `completed` datetime DEFAULT NULL,
  `description` text,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;


CREATE TABLE `trac_enum` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `type` varchar(30) DEFAULT NULL,
  `name` varchar(50) DEFAULT NULL,
  `value` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

Launch a database GUI or command line client, locate database that was created during Cubi installation wizard. Say the cubi database name is “cubi”. Run the document create table statement under “cubi” database.

Comment by chitwar...@gmail.com, Oct 9, 2011

this is really helpful since i have openbiz in my head and cubi but no freaking idea still how to build anything with it,i mean i see the palace but no gates or even roads to the place

Comment by thekenya...@gmail.com, Aug 19, 2012

Very helpful resource.

Comment by yukinosa...@gmail.com, Jan 25, 2015

yup

► Sign in to add a comment