Implementation Guide

Sapientia Canteen MVC

Roland Bogosi

Computer Engineering III

The website uses a custom MVC engine written from scratch, which implements all the basic functionalities of said architecture.

All requests that are not mapped to a *file*, *folder* or *link* by the webserver, is rewritten to the *index.php* file, where the MVC class initializes itself, parses the request URI, and tries to map the request to a corresponding controller. Special error handlers are set up in case something goes wrong. If things do not go south, though, the *ControllerBase*-implementing controller will have access to all the basic functionalities of an MVC system, such as the Models (and the underlying Database with Caching), the View and possibly other Controllers should it be required to change the flow.

The Model part of the system, implemented by the *ModelBase* abstract class, provides highly advanced reflection capabilities to the implementing class, such as that a fully functioning model can be achieved with little or even no meta-data, other than the member fields of the implementing model. Other extra features, such as SQL table generation are also available, and as a result, a programmer may fire up a random class, call *create()*, at which point a *CREATE TABLE* request will be issued, and the table is now ready to be worked with.

The View part of the system, implemented by the *View* class, provides primitive but highly performant and low-memory rendering to the other components of the system. Additional functions can be performed through this class, such as user redirection.

The Controller part of the system, implemented by the *ControllerBase* class, provides full access to other components of the MVC system, and light aspect-oriented features, such as controller *enter()*/*exit()* methods for initialization, sanity and security checking.

The class that ties it all together is implemented under the name *MVC*, and is responsible for holding references to all the components, holding miscellaneous meta-data and routing information, which will be used to route the incoming request to a controller.

The administration interface is fully based upon the *CrudHelper* controller, which provides automatic full dynamic Create/Read/Update/Delete operations support for any class implementing the *ModelBase* class.

The user management functions are provides by the *UserMgmt* controller. A user may register, login temporarily/permanently and request a password reset. Security best practices are followed, and all known attack vectors have been mitigated.

The payment functions are handled via a 3rd-party webservice called *Stripe*, in the *Canteen* controller. Transactions are tracked in the local database. Security best practices are followed in this case as well, the website functions fully under HTTPS and the credit card information is never sent to our server, only to *Stripe*, and secure tokens are used for further communication.

Other features, such as menu listing and reservations are running from under the *Canteen* controller, using all the components of the MVC system in harmony.