

## Concerto - Your data, where and how you want it

### Abstract:

People want data. My goal, over the summer, is to provide an open source data presentation and storage system written in PHP, and provide the foundation for future improvements and integrations. There are dozens of commercial solutions on the market that will take any given data or media, and display it in a particular format; whether that be on a website like a CMS system, on a television for a larger audience, sent to a mobile phone, or a printed hardcopy. There are no open source systems which effectively address the presentation and storage needs of all these different options in one simple to use, manage, and extend product.

For the past year I've been working with a team on a product we call "Concerto", a digital signage system initially for use at my college, RPI, where anyone can submit content to feeds (categories) which have moderators. On the other end of the system a screen, or portal into the system, subscribes to and weights those feeds for display on their individual outlet (be it TV screen, web site, screen saver, etc). The goal has always been to provide an easy way for people to get the data they want where they want it, and provide a unified outlet for people to distribute their content.

To date the project has been much focused on displaying content on large TVs/monitors. At each site we setup a bare-bones computer running of CF card to boot a custom compiled Debian install to open up a Firefox browser pointed to our display page. But this is only one portal into the content the system houses, and over the summer I'd like to create an API to allow anyone to peek in and see what's inside.

---

### Project Description:

I understand submitting my project to Google as a mentoring organization is kind of a crazy idea, but I'll be working on this regardless of if I get accepted as an official project or not. I also feel Google may be the best place to seek a mentor experiencing in creating an API that provides tons of functionality to developers in an easy to use fashion.

Concerto, as the system is appropriately used, has lots of potential applications and extensions in today's data centered world. New ways of accessing content are popping up daily. From the internet, to cell phones, televisions to media extenders to Chumby's... everywhere you look there are more and more ways to be connected. The problem is a lack of unification on the presentation layer. The internet is great for accessing data, but the standard website provides a poor implantation of personalized data presentation, or presentation to the masses. Enter Concerto.

A company could buy a \$5,000 digital signage system and install lots of large displays around their site, or you could install a free open source tool to do it all, and more. People wouldn't have to turn around to look at a big TV if they didn't want to, they could visit a website. Not at your computer, pull out a cell phone or glance over at the scrolling LED message sign. I see a day where people go to one place to aggregate their content, knowing it can follow them anywhere onto any presentation type. This is

possible with Concerto. The same system could be used in a home environment, imagine getting ready in the morning while watching TV and your RSS feeds are scrolling across the bottom. Maybe a note from work pops up; you're 10 minutes late for a meeting. To make this all happen, Concerto needs a strong and easy to use API. If things are hard, few people are going to understand how to implement it, and that's not very nice. To do this, an http based XML API would be written to allow system to perform all the functions the suite offers, starting with the rendering of content.

Development of the API would proceed similar to this:

1. Develop an authentication scheme to allow developers and their applications to securely connect and post/retrieve data to and from the system. This would likely involve a "secret" key, and an https connection.
2. Create an API layer on top of the current class structure. (All interactions with the database and objects are handled by classes). Developers shouldn't have direct access to the classes and their internal functions like the SQL queries, but the layer should provide all the functionality needed to CRUD all public data in a secure and simple fashion.
3. Expose the API layer via the authentication scheme. This would probably entail writing a client library to easily form and handle responses from the sever side API layer.
4. Document it so all can use it.

This probably sounds like a very daunting task that might take a team of full time programmers several months to release, and that probably isn't very far from the truth. However I feel that over the duration of a summer significant progress could be made on the system.

I would propose the following criteria to judge the projects progress:

- Authentication scheme in place, data can be securely sent to and from the sever on a per developer basis
- Core functionality to retrieve content from the system in a secure fashion is exposed via API
- Client libraries written for PHP to authenticate and retrieve content in a secure fashion.
- Documentation of the client library, and the functionality exposed via API sufficient to allow others to create client libraries and interact with the API.

If this was a waste of your time I'm sorry. Maybe you enjoy playing hide and seek with the data you want, or maybe you enjoy others controlling exactly what you see when you turn on the TV.

Thanks for reading and I appreciate your consideration,

Brian Michalski

P.S. If you do happen to be interested in sponsoring this project, but are confused about Concerto I'd be more than happy to answer any and all questions you can dream up. Shoot me an email or something.