

Pinger-DB Implementation details

At a moment Pinger-DB is implemented on three different packages.

1. (DAO) Data Access Objects
2. Data Processing
3. Visualization

Data Access Objects

Each query for the database has its own Module. E.g. If you want to query for MetalIDs from pinger database. There is a separate module `PINGER::Database::MySQL::Meta` and for pings `PINGER::Database::MySQL::Ping`, which are both extended from `PINGER::Database::MySQL`. `MySQL` module require all the global configuration properties to connect to database. `MySQL` is responsible for connecting and disconnecting from the database. It accept the configuration parameter in a standard format defined using `PINGER::Config` module. The most important module in this package is `PINGER::Factory::PingFactory`, to which you only specify source and destination pair and time interval, it then handle all possible

Data Processing

When data is fetched from the database using DAO, the data is processed for different errors. This package mainly contains for fetching the data in a specific formats either in `Pinger::Tools::Ping` objects or in a JSON format. The modules that handle such task will be under heirarchy `PINGER::Canvas`.

Visualization

Currently, the visualization is based on Simile Exhibit framework. The framework accepts data in a specific JSON format. The files responsible for visualization are in `pinger-db/trunk/bin/*` directoty. In order to test the visualization you need to set apache `cgi-bin` directry to `pinger-db/trunk/bin`.

Checkout Source from SVN

Use this following command on a machine that is on SLAC domain.

```
svn co file:///afs/slac.stanford.edu/g/scs/net/netmon/repo/svn/pinger-db
```

After checking out the code, set the `PERL5LIB` environment variable to the `pinger-db/trunk/lib`. you will also require the config file having connection parameters to database which you will find at `/afs/slac.stanford.edu/u/sg/mak/config`.

Examples

After checking out the code, you will find code snippets example for getting to know how to use `PingFactory` and database modules in a `pinger-db/trunk/unit-tests` directory.