Table of Contents

Quick Start Up Guide for PanDA Web Platform Application Developer

Introduction

The new Panda Web Platform was developed to back the Panda Monitor Web server that is more maintainable, supports the JSON/JQuery architecture, is easily extensible, and integrates well with other ADC monitoring tools and components

- Supported job archive databases and the arbitrary data source (for example, the local 'root' file and an external Web Service).
- Functional modules are included automatically and dynamically. There is no hard coding of functional components or user interface elements. pmModules/modname.py is used to interpret URLs of the form http://baseurl/modname/?param1=1&param2=2
- Standard module routines publish the data to be sent to the client. The module can specify the various roles of the data it publishes via "publish" method
- The data the module publishes is interpreted, compressed and sent to the client / browser according the the data role defined as parameter of the publish method and the client type / capability. so monitor modules can serve as either web page builders, or json data providers or the rendering function providers or all above

The goal of this topic is to assist the PandDA developers to create he/ her first HelloWord−level custom Web application within the new Panda Web Platform

Three Components

To introduce the new Web application backed by the "Panda Web Platform" one needs to design and provide up to 3 software components, one mandatory and 2 optional

1. **Python module** to publish the application Web API and the custom Web Content *(OO programming with "python" language is required)*
2. (opt) **JavaScript function** to render the published Web Content *(programming with JavaScript, JQuery, and HTML are required)*
3. (opt) **Data Access Layer** to fetch the external data used to generate the Web Content *(the required skill is defined by the nature of the data. For example, to fetch the data from Db one needs to know SQL)*

How to Start

1. Choose the working host . There are three host machines available for the PAS/BNL PanDA developers :
   http://pandamon.atlascloud.org
   http://pandamon-new.cern.ch

1. Ask either S.Baranov to register you as "module developer" and provide the developer account.
2. Upon registration make sure your version is working
   http://pandamon-new.cern.ch/~gayazov
3. Create your Web application from the existing example (recommended) or from the scratch (is not recommended)
4. Deploy your application
5. Test it.

Create Your Own Context and Web Application

The "platform" application is an instance of the python class derived from the pmModule class (See Appx A. for details). In addition the brand-new application should satisfy the constrain, namely,

The name of the python source file and the name of the python class must be the same

Module "Objectives"

Web application is the subclass of pmModule class. It should (see Appendix C and Appendix F)

- **Publish** one of its* own method* as "UI API" using **pmModule.publishUI** method (optional)
- **Create** some Python **object** and and **publish** it as the "Page Content" using **pmModule.publish**; method
- Use (optional) **pmModule.publish** method to to publish the custom **JavaScript** function to render the "Page Content"

Create the First Web Application

- **cd $PANDAROOT/panda/pandamon/pmModules**
- Optional: Create a new subdirectory:
  mkdir stavro
- Copy the file hellojs.py and give it the new name
  cp hellojs.py stavro/histavro.py
- Edit the file: **stavro/histavro.py**
- Find there the line:
  class hellojs(pmModule):
  and replace it with:
  class histavro(pmModule):

Save the file. Ensure the class name **"histavro"** match the file name **"histavro.py"**

Deploy It!

To (re-) deploy the brand-new application one needs to change the framework 'wsgi' script. The simplest way to do this is to apply the touch shell command. Just do

touch $PANDAROOT/pandamon.wsgi

That's

How to Test

As soon as you deploy your brand new application you should be able to access it via the Web Browser and from CLI

For example:

http://pandamon-new.cern.ch/~gayazov/stavro/histavro

curl --compressed 'http://pandamon-new.cern.ch/~gayazov/stavro/histavro'

To change the code just edit its source code with your favorite text editor and re-deploy again

Next

The "hello" application is pretty useless unless it can access the real data from PandaDB to publish. The page How to access the Panda data using TaskBuffer API

Major updates:
-- ValeriFine - 13-Feb-2013

Responsible: ValeriFine

Never reviewed

This topic: PanDA > PandaPlatformDevelopers
Topic revision: r10 - 2014-02-06 - ValeriFine