APRIVA®

Confidential and Proprietary

Apriva PHP Sample App Getting Started Guide

ECLIPSE

Document Version: 1.1

Last Updated: May 15, 2018

This document is the Apriva PHP Sample App Getting Started Guide.

Rights to the contents of this document

Modifications to the contents of this document

APRIVA reserves the right, at any time and without notice, to change these materials or any of the functions, features or specifications of any of the software described herein. APRIVA shall have no warranty obligation with respect to these materials or the software described herein, except as provided in APRIVA's software license agreement with an authorized licensee.

Trademarks

APRIVA ® is a registered trademark of APRIVA
APRIVATalk™ is a trademark of APRIVA

Comments

Please send your comments, questions, or suggestions about this document to:

Apriva, LLC
ATTN: Apriva Integration Services
8501 N. Scottsdale Road, Suite 110
Scottsdale, AZ 85253
(480) 421-1200
integration@apriva.com

OVERVIEW

Apriva Integration Services provide APIs and SDKs to securely connect your POS solutions to 35 North American payment processors on nearly any platform. Apriva is empowering developers to create applications that can sell anything, anywhere.

Apriva's proven technology has been rigorously tested and verified by the U.S. intelligence community, Department of Defense, and the leading organizations and standards bodies in the payments industry including Visa®, MasterCard®, and the PCI Security Standards Council. Apriva products have been unlocking the potential of mobile devices for more than a decade, and can provide Independent Software Vendors (ISVs) with unrivaled features and flexibility in payment processing.

Specific to POS solutions written in PHP server-side scripting language, Apriva offers the Apriva PHP sample application to programmatically integrate with the Apriva Gateway™ using XML.

INSTALL THE APRIVADEVELOPERCERTIFICATE

Apriva first credentials you to access the Developer Portal, BackOffice™, and the development/testing environment of the Apriva Gateway. You then develop and test your integration in that environment, with Apriva providing integration support as needed.

Once your POS solution is fully-integrated and tested, you schedule Apriva Certification where Apriva sends you a connectivity script to test your connection and transaction processing. Upon successful certification testing, Apriva certifies your solution and grants you access to the production environment of the Apriva Gateway.

The Apriva Gateway requires a client certificate for accessing its end points. As part of the sample application, there is a development certificate that can be used to get familiar with the process. The Apriva Integration team will provide you with your own development certificate as part of the development environment credentials.

NOTE

In the PHP sample application, the AprivaDeveloper certificate is loaded automatically, so you need not take any additional steps to acquire this certificate.
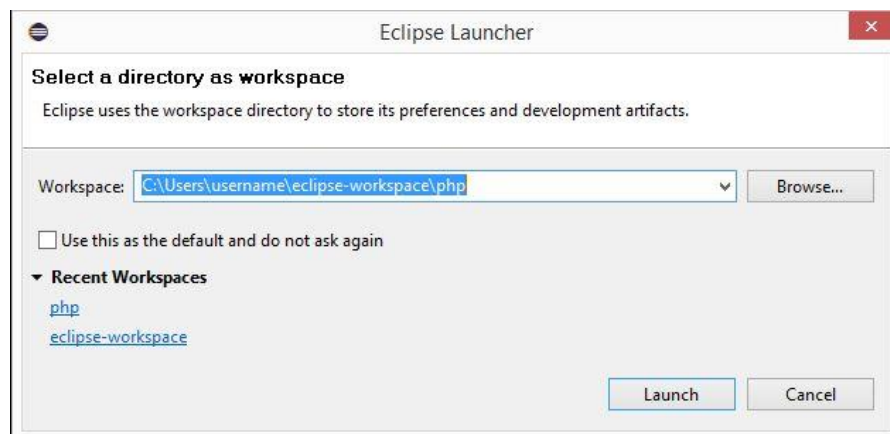
IMPORT THE PROJECT INTO ECLIPSE

The Java sample application showcases the Apriva Gateway API's features. For this example, we used the **Eclipse IDE, Luna Service Release 1**:
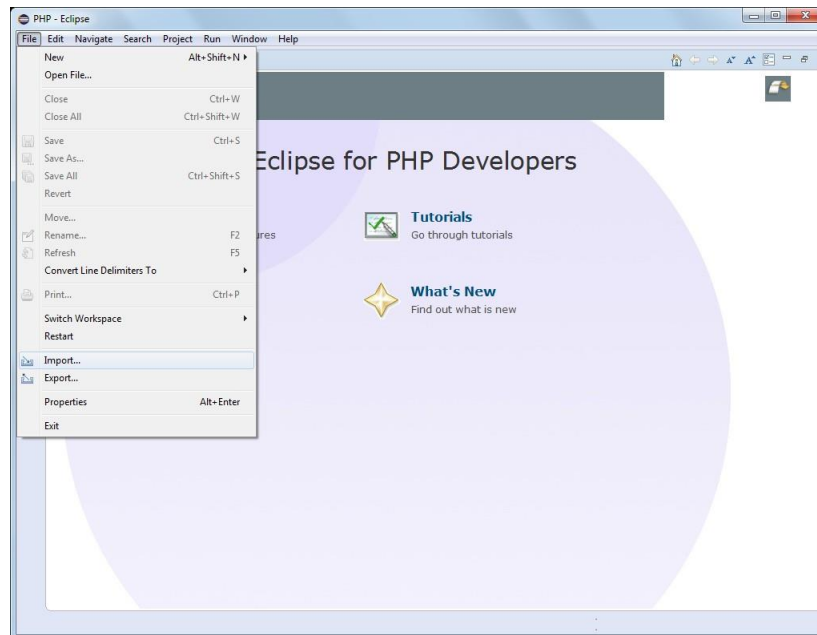
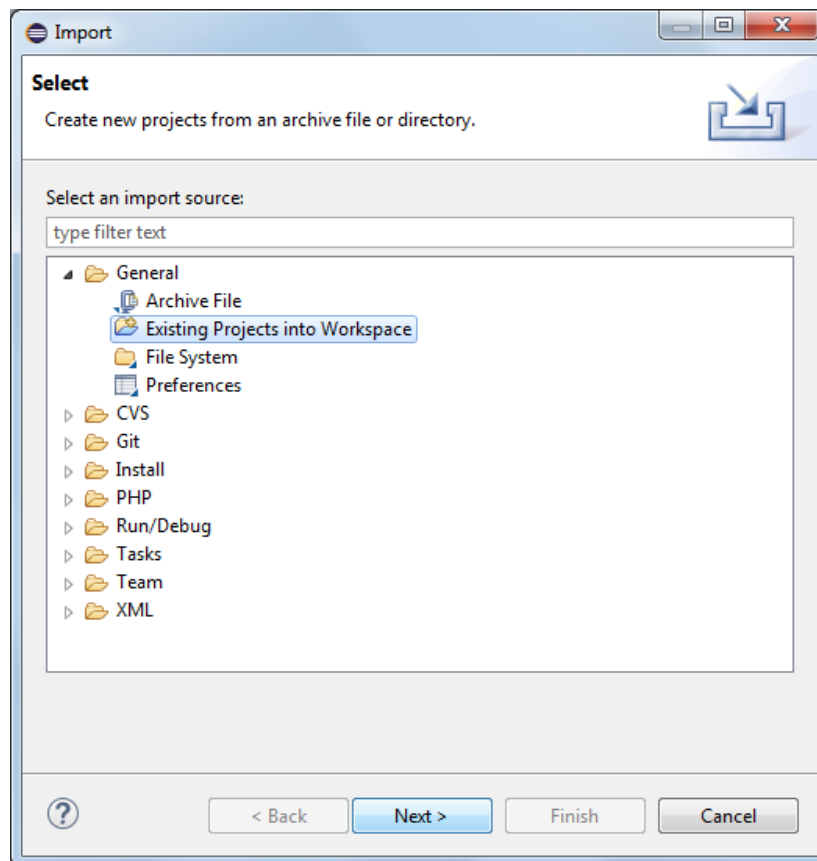To launch the Java sample application, follow these steps:

1. Start Eclipse. If the workspace Launcher window pops up, select a workspace. For this example, we used "**php**". Click the **OK** button.
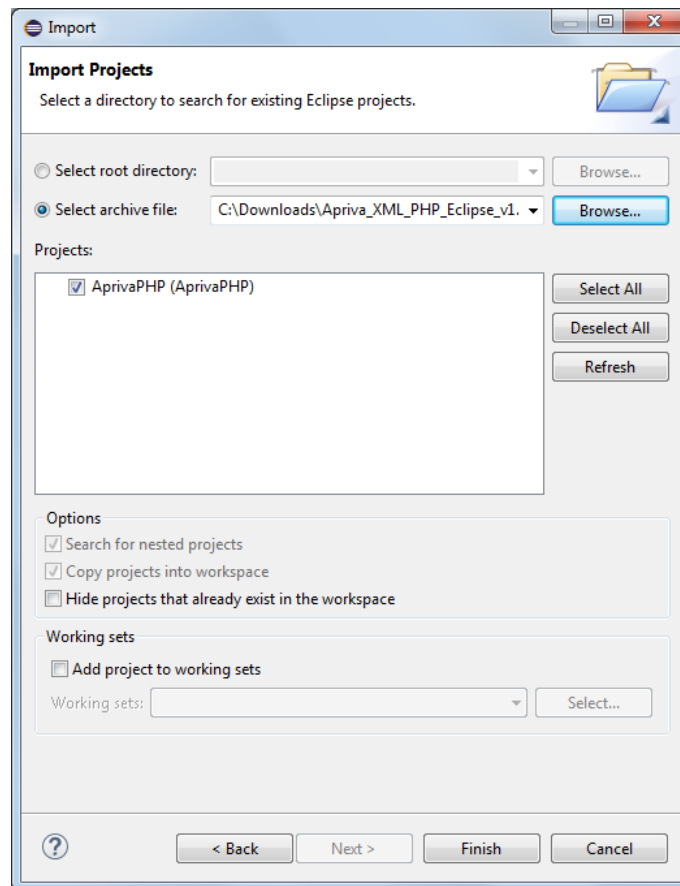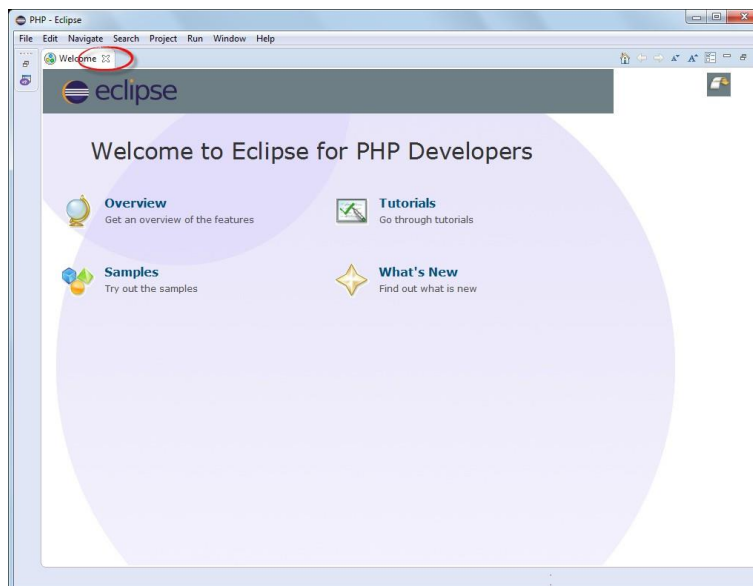
2. Select File → **Import…**

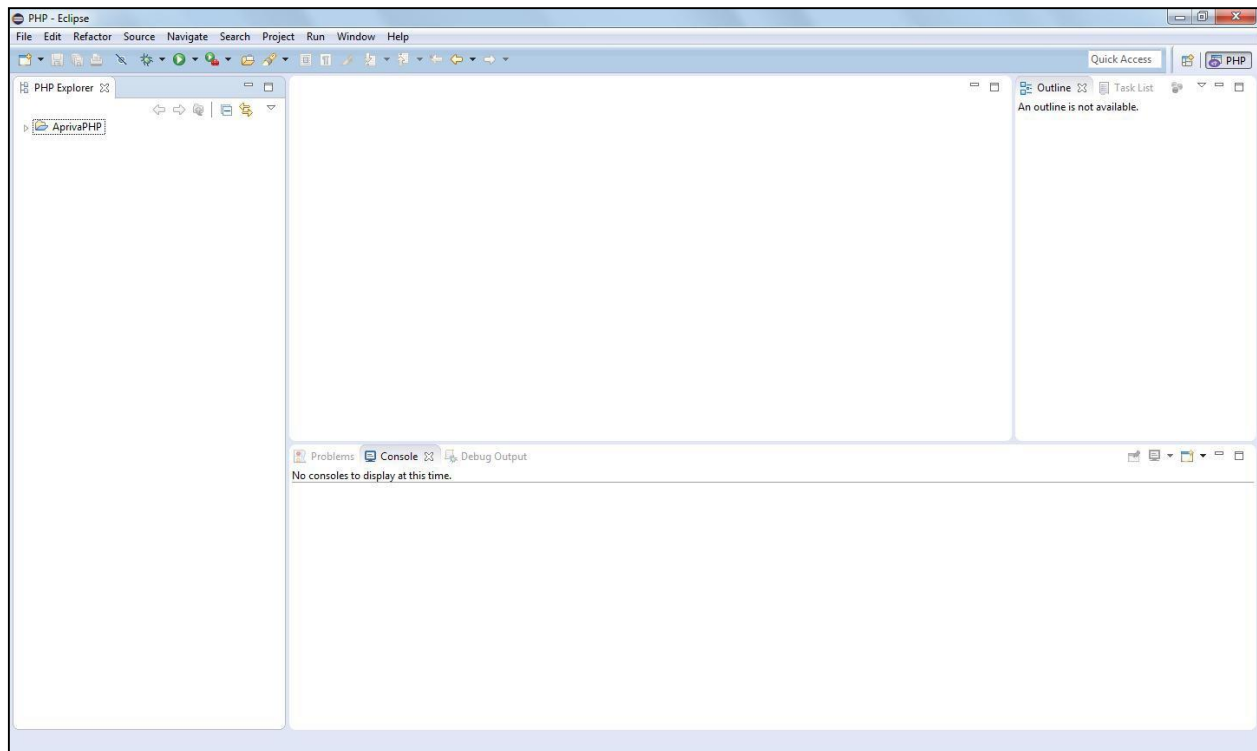**3.** Expand "**General**", select "**Existing Projects into Workspace**", and click the **Next** button.

4. Select the **Select archive file** radio button . . .

5. And browse to locate the "**Apriva_XML_Java_Eclipse_v1.1.zip**" archive, then click the **Finish** button. The project will be imported into the current workspace.

6. Close the "**Welcome**" Tab.

7. The "**Package Explorer**" view will now be displayed.

SET UP THE PHP ENVIRONMENT

Once you have the project imported into Eclipse, you must set up the PHP environment:

1. Ensure that you have PHP installed, and confirm that the **php.ini** file contains "**extension=php_openssl.dll**" (see below):

```
894   ;extension=php_mysqli.dll
895   ;extension=php_oci8_12c.dll   ; Use with Oracle Database 12c Instant Client
896   extension=php_openssl.dll
897   ;extension=php_pdo_firebird.dll
898   ;extension=php_pdo_mysql.dll
```
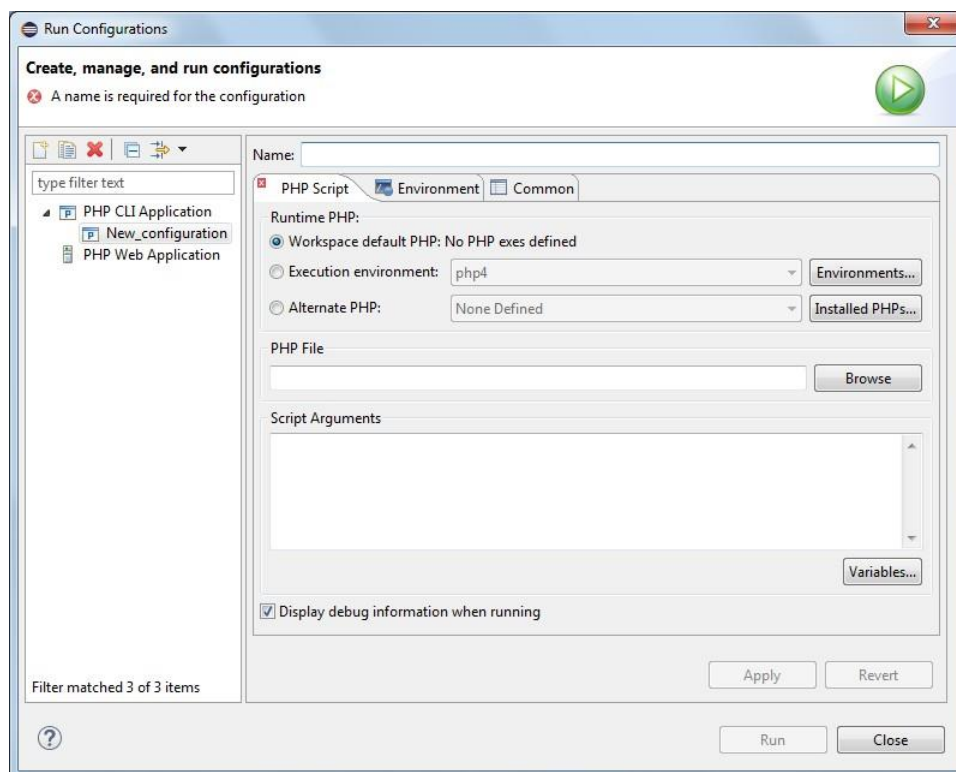
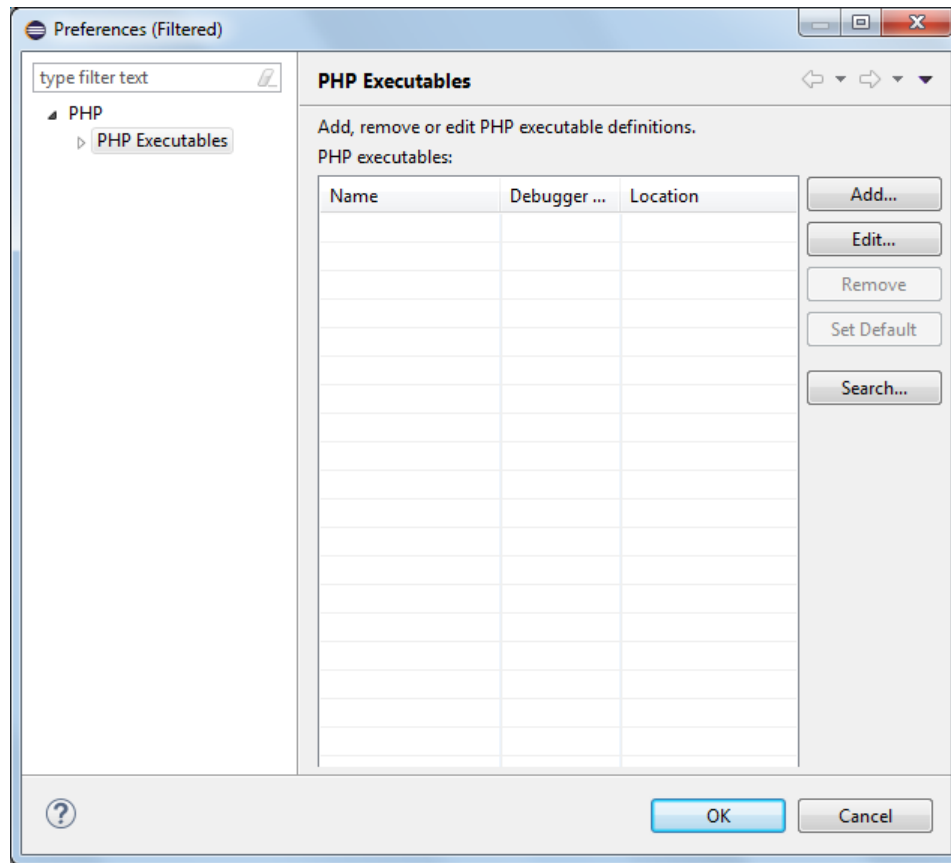2. Select Run → **Run Configurations...** from the main window menu:
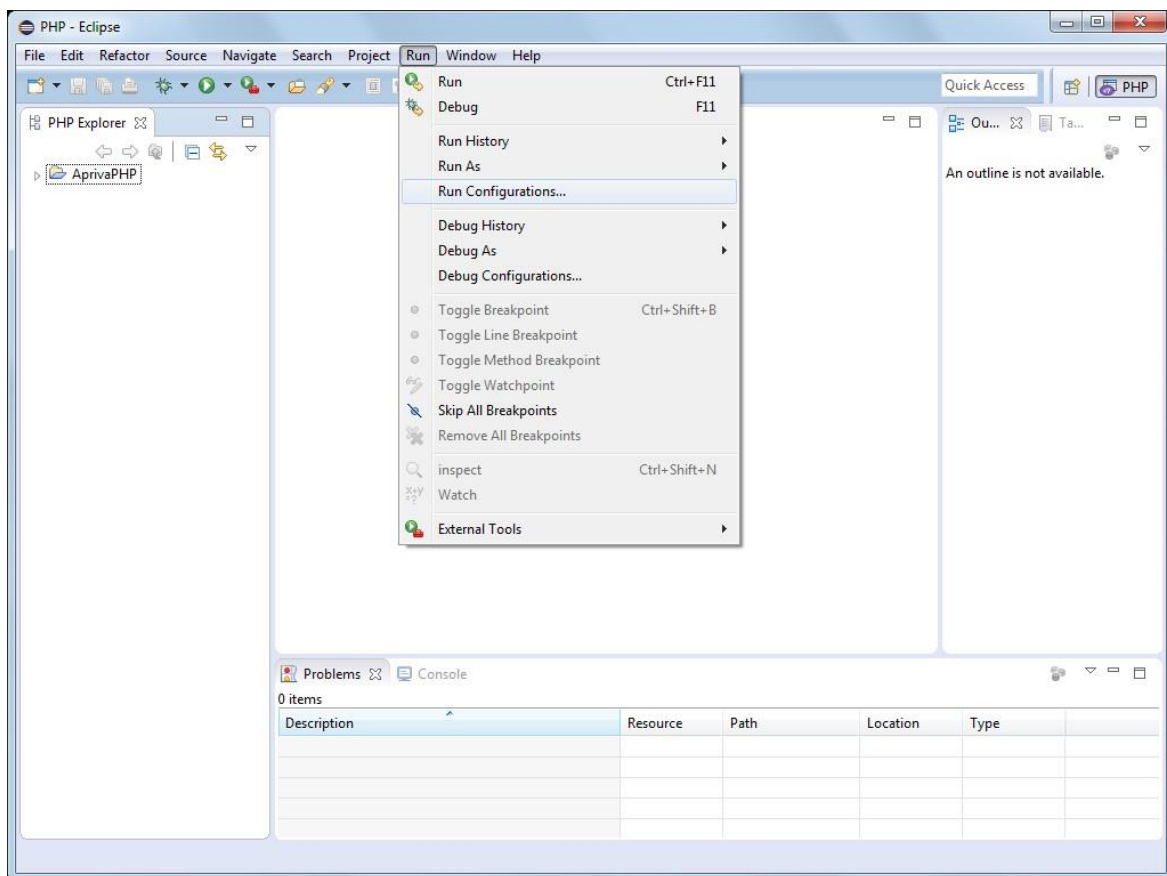
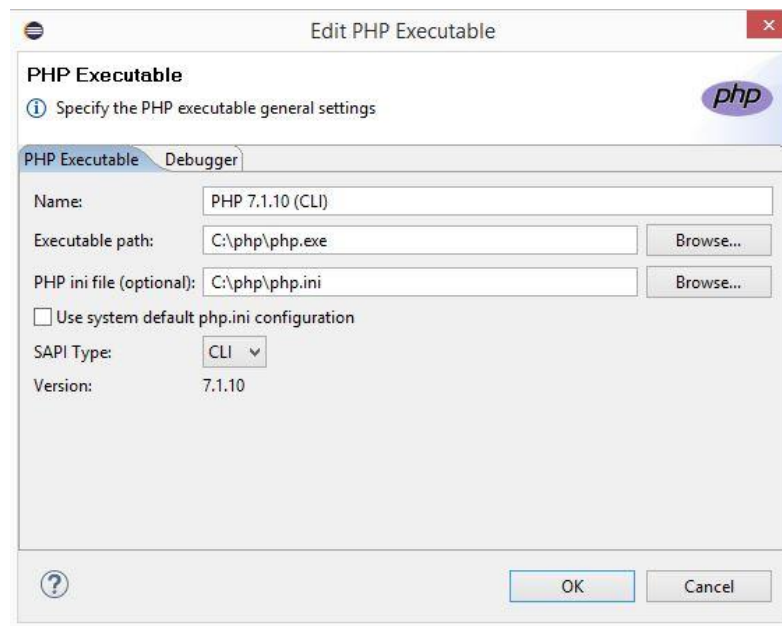3. The **Run Configurations** window will open . . .
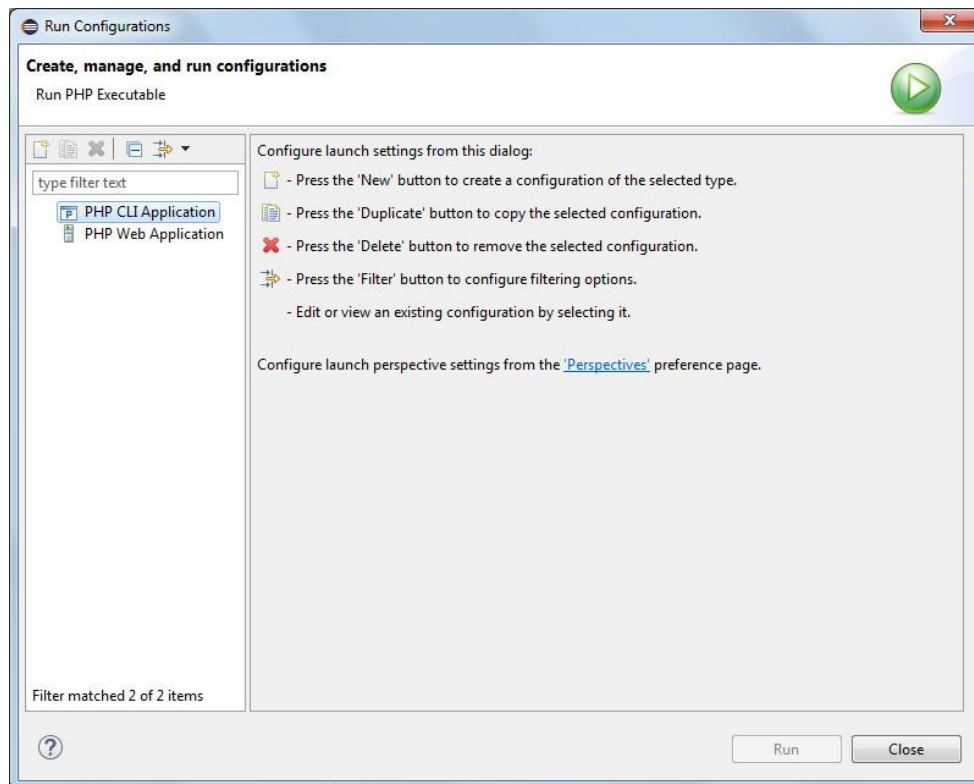
4. Double-click the "**PHP CLI Application**" and the Configuration window will appear.

5. Click the **Installed PHPs** button and the "**PHP Executables**" window will appear.


6. Click the **Add…** button and the "**Add PHP Executable**" window will appear.

7. Click the **Browse** button associated to the right of the "**Executable Path**" and select your local "PHP.exe" file location.

8. The "PHP ini file" field should automatically complete. Enter a name for the top "**Name**" text field. For example, PHP5.5.12 — your text fields should resemble this:
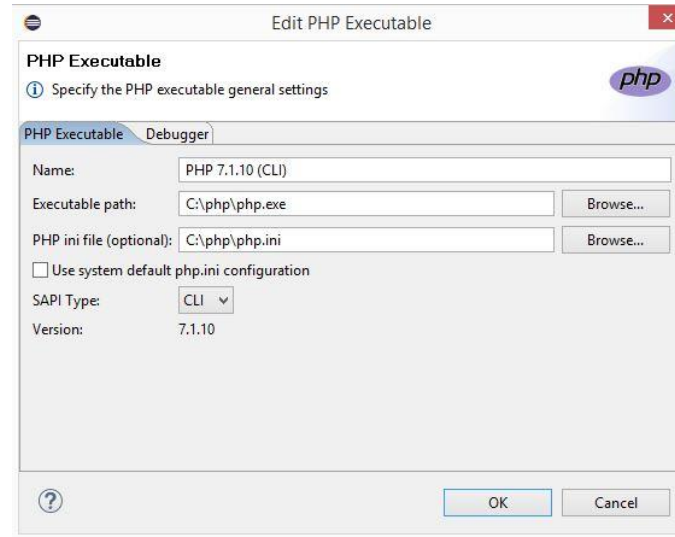
9. Click the **OK** button, returning you to the "**Run Configurations**" window. In the top "**Name**" field, enter "**onSubmit**" as the executable name.
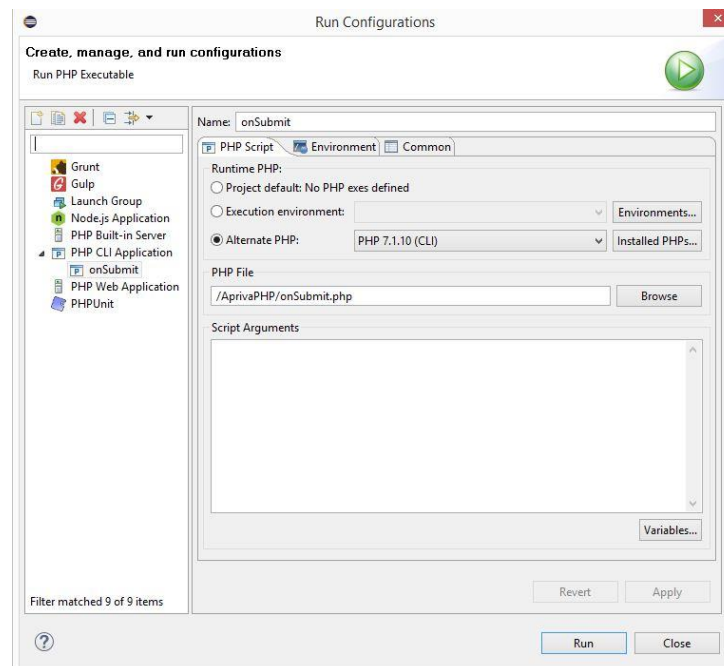

10. Click the "**Alternate PHP**" radio button, where your PHP setup will be selected. Enter "**/AprivaPHP/onSubmit.php**" in the "**PHP File**" field and uncheck the "**Display debug information when running**" box.

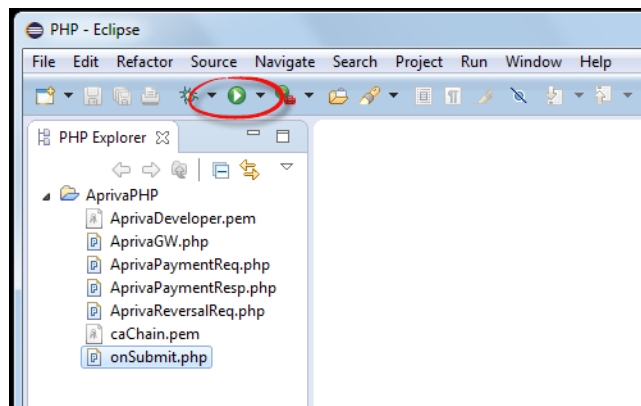11. Click the **Apply** button and then click the **Close** button.

EXECUTE THE CODE

To execute your new PHP sample application code, follow these steps:

1. Under the "PHP Explorer" view, expand the "AprivaPHP" directory. Left-click the "onSubmit.php" source file so it is selected.
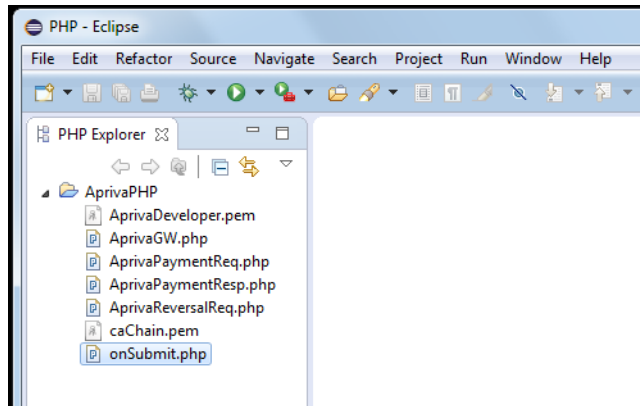
2. Click the **Run** button to execute the PHP code.

3. The console window will display your XML requests and responses: