

## Seller Hub

### (Web based private API for application integration)

**Sneha Ambavale<sup>1</sup>, Dipika Thakare<sup>2</sup>, Mehwish Khan<sup>3</sup>, Assist. Prof Nilesh Rathod<sup>4</sup>**

<sup>1</sup>Student, IT, Rajiv Gandhi institute Of Technology, Maharashtra, India

<sup>2</sup>Student, IT, Rajiv Gandhi institute Of Technology, Maharashtra, India

<sup>3</sup>Student, IT, Rajiv Gandhi institute Of Technology, Maharashtra, India

<sup>4</sup>Assistant Professor, IT, Rajiv Gandhi institute Of Technology, Maharashtra, India

\*\*\*

**Abstract**— In order to improve business by automating business processes that span these isolated systems, the Enterprise needs to integrate the isolated systems and applications with Enterprise Application Integration (EAI), which would be linking diverse systems and applications of the enterprise to enable the enterprise to adapt to the dynamic business environment. Whereas, with traditional EAI solution, it is difficult to efficiently link the different proprietary applications and data sources, and difficult to enable the system to rapidly identify and respond to changes in the dynamic business environment. The issue of communication between multiple applications developed on same or different platforms is paramount in any organizations as almost all major businesses cannot be done without getting online these days. Now it is need of every organization to develop applications for their business which can easily communicate with each other for their working. Private API is the great framework for exposing your data and service to different-different devices. Moreover API is open source an ideal platform for building REST. It uses the full features of HTTP and you don't need to define any extra configuration settings for different devices. Instead of every application communicating with each other, every application would only communicate with the API. So that API keeps all the data and application.

**Key words**— API, Web services, EAI

### 1. INTRODUCTION

Most of the enterprises might use some kinds of isolated IT systems, such as Enterprise Resource Planning (ERP) systems, Customer Relationship Management (CRM) systems, Supply Chain Management (SCM) systems, Office Automation (OA) suites and so on, to automate distinct business practices. These systems often coexist with a wide variety of in-house applications. The traditional EAI technologies and the bottlenecks for applications integration, also elaborates the Web Services technology and the advantages suited for applications integration. Therefore, the model of EAI with Web Services (EAIWS) is proposed to efficiently implement the integration of

diverse applications and systems within or between the enterprises. With EAIWS, provides an example to explain how the application system works. The approach is demonstrated by building a real-world application for EAI in the financial services domain. The enterprise needs to integrate the isolated systems and applications with Enterprise Application Integration (EAI), for improving business by automating business processes, which would be linking diverse systems and applications of the enterprise. The development of Web Services technologies which is based on open standards (such as XML, SOAP, REST), easy to use and widely supported and deployed, provides a better integration solution in our approach, we consider to use Web Services technologies for EAI. For the development of Private API we are going to use Symfony Framework. The Online transactions will take place by the PAYPAL Integration and also the security to users will be provided by OAUTH2.0 protocols.

### 2. IMPLEMENTATION

The objective of designing a new system for EAI with web services system is to improve the communication between multiple application and time efficient system over that of the traditional EAI system.

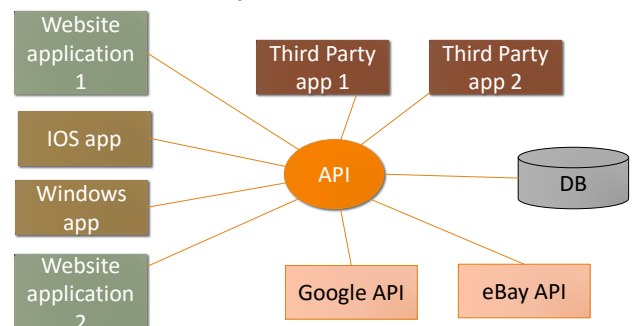
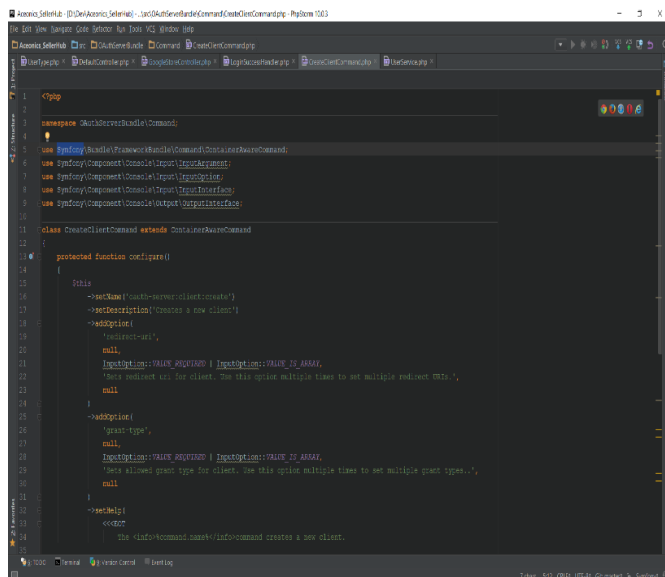


Fig 1 .Block Diagram

The following are the methods that will be used in our project:

- Our proposed system will have REST based private API to add/edit/delete/read the products.
- To edit/add/delete operation this API would call API's of Google, EBay and perform the respective operations.
- Users of the API will have normal user role & administrator role. Authentication would be required for users to use the API.
- One web based applications to call this API and demonstrate API features with reports of the API usage to administrate through the graphs. On desktop software applications to call this API and demonstrate API features with reports of the API usage to administrate through the graphs.
- On desktop software applications to call this API and demonstrate API features with reports of the API usage to administrate through the graphs.



```

1  <?php
2  namespace SellerHub\Vendor\Commands;
3  use Symfony\Component\Console\Command\Command;
4  use Symfony\Component\Console\Input\InputArgument;
5  use Symfony\Component\Console\Input\InputOption;
6  use Symfony\Component\Console\Output\OutputInterface;
7
8  class CreateClientCommand extends Command implements BaseCommand
9
10     protected function configure()
11     {
12         $this
13             =>setName('create-client')
14             =>setDescription('Create a new client')
15             =>addArgument('client-id', InputArgument::REQUIRED, 'Client ID')
16             =>addArgument('secret', InputArgument::REQUIRED, 'Secret')
17             =>addArgument('grant-type', InputArgument::REQUIRED, 'Grant type')
18             =>addOption('client-type', InputOption::VALUE_NONE, 'Client type for client, use this option multiple times to set multiple client types.')
19             =>addOption('secret-type', InputOption::VALUE_NONE, 'Secret type for secret, use this option multiple times to set multiple secret types.')
20             =>setHelp('
21                 See "/>

```

Fig 2:Creating Client command

Fig 2 tells about the creating the client command means how the user is going to interact with the logging system. Client will makes use of the Oauth protocol for the security purpose.

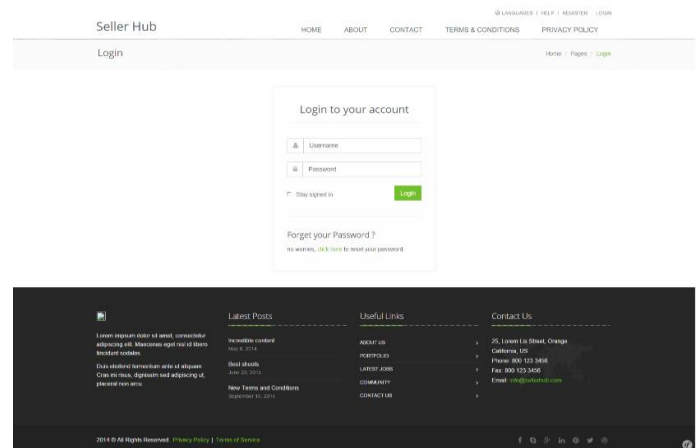
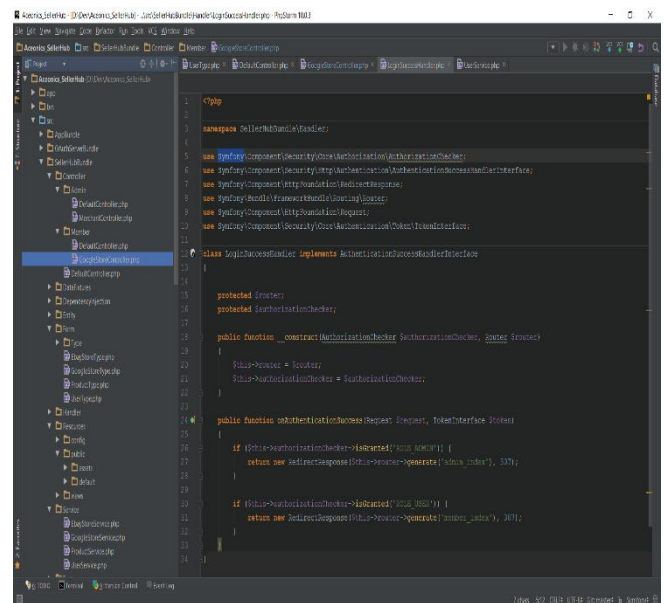


Fig 3:Logging in UI

Fig 3 shows how the user will logging into the system and interact with API to CRUD the products. Logging will authorize only the authenticate user to access the administrator privileges.



```

1  <?php
2  namespace SellerHub\Vendor\Handlers;
3
4  use Symfony\Component\Security\Core\Authentication\AuthenticationChecker;
5  use Symfony\Component\Security\Http\Authentication\AuthenticationSuccessHandlerInterface;
6  use Symfony\Component\HttpFoundation\RedirectResponse;
7  use Symfony\Component\HttpFoundation\Session\Session;
8  use Symfony\Component\HttpFoundation\Request;
9  use Symfony\Component\Security\Core\Authentication\Token\TokenInterface;
10
11 class LoginSuccessHandler implements AuthenticationSuccessHandlerInterface
12 {
13     protected $request;
14     protected $authenticationChecker;
15
16     public function __construct(AuthenticationChecker $authenticationChecker, Request $request)
17     {
18         $this->request = $request;
19         $this->authenticationChecker = $authenticationChecker;
20     }
21
22     public function onAuthenticationSuccess(Request $request, TokenInterface $token)
23     {
24         if ($this->authenticationChecker->isAuthenticated($token, $request)) {
25             return new RedirectResponse($this->request->generate('admin_index'), 302);
26         }
27         if ($this->authenticationChecker->isAuthenticated($token, $request)) {
28             return new RedirectResponse($this->request->generate('admin_index'), 302);
29         }
30     }
31 }

```

Fig 4:Login success handler

Fig 4 will be handling the logging processs. It will limit the unauthorized user from accessing it. Once the user will get the right to access the UI. If user is authorize then the successful login is done.

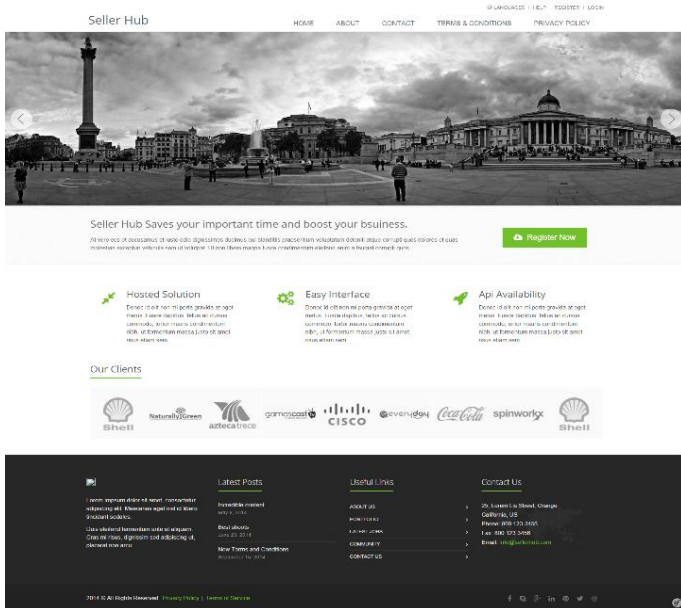


Fig 5: Seller Hub UI

Fig 5 shows the Home page of the Seller Hub. It having navigation bar on top. For new user it will register them by clicking on register button. Role to the user will be assign by the seller hub team.

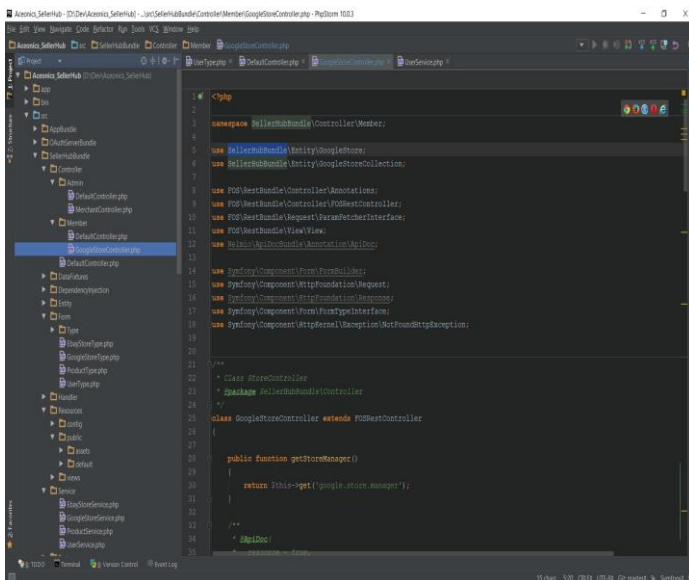


Fig 6: Google store controller

Fig 6 shows the code for google store operations. This will control all the operations by of the Google store. If user wats to addhis products on the Google shopping the all the calls to the CRUD store will be handle by this code. Then afterwords by the Google API.

### 3. ADVANTAGES

1. Based on REST protocol instead of SOAP which helps to easily add or link a new legacy application.

2. It is easy to integrate applications across various platforms, operating system and programming language.

3. All of systems communicate and exchange data through the API.

4. Multiple applications can communicate with API at the same time.

5. Simple and Flexible.

### 4. CONCLUSION

All kind of applications developed on multiple platforms like Windows, Android and iOS access Web API to communicate. This solves the major maintenance problem by making things easily manageable, which results in saving companies both time and money. Instead of every application communicating with each other, every application would only communicate with the API. So that API keeps all the data and application.

### Acknowledgment

This research paper is made possible through the help and support from everyone, including: parents, teachers, family, friends, and in essence, all sentient beings. Especially, please allow me to dedicate my acknowledgment of gratitude toward the following significant advisors and contributors: First and foremost, I would like to thank Prof. Nilesh Rathod for his most support and encouragement. He kindly read my paper and offered invaluable detailed advices on grammar, organization, and the theme of the paper. Second, I would like to thank our H.O.D Prof. Dilip Dalgade to read my thesis and to provide valuable advices, Finally, I sincerely thank to my parents, family, and friends, who provide the advice and financial support. The product of this research paper would not be possible without all of them.

### References

- [1] Jeff Pinkston, eAI Journal, 2001,8, 48-52
- [2] Jeffrey .C, Web Services Architecture Usage Scenarios W3C Working Draft, <http://www.w3.org/TR/2002/WD-ws-arch-scenarios-20020730/>, July.30 2002
- [3] Rick Kuzyk, eAI Journal, 2002, 4, 23-25
- [4] F.Leymann, IBM System Journal, 2002,41(2), 31-36
- [5] Rosen and J.Boak, eAI Journal, 2002, 1, 43-46
- [6] Scott Bluman, XML-enabled Enterprise Application Integration(XAI):Fundamental for B2B integration, <http://xml.e-centre.org.uk/download/XML-enabled/EAI.pd,20>