# Jaspersoft APIs

## Integrating BI with your Applications

### Community and Professional Editions


**Jaspersoft Headquarters:**
539 Bryant Street, Suite 100
San Francisco, CA 94107, USA

www.jaspersoft.com

Email: sales@Jaspersoft.com

Telephone:
+1 888.399.2199 or
+1 415.348.2380

**Jaspersoft Europe, Middle East, and Africa:**
Digital Court
Rainsford Street
The Digital Hub
Dublin 8, Ireland

Email: sales-emea@jaspersoft.com

Telephone:
Germany +49 30 8939 1934
UK + 44 207 193 9321
France + 33 970 446 126
Italy + 33 970 446 126
Spain + 33 970 446 126
Poland + 48 22 219 6087
Ireland + 353 1 443 4700
Switzerland + 41 44 586 76 99
Sweden + 46 85 19 71 245

# Contents

# Introduction

Almost every commercially viable business application offers built-in reporting or analytics. Too often, however, the business intelligence (BI) features included with these applications fail to meet customer expectations. At the same time, product managers and developers become increasingly frustrated with home-grown and legacy systems that are far too expensive to develop, manage, and maintain.

Jaspersoft solves these problems, so that reporting and analytics can truly enhance the quality and value of all applications through seamless integration. Jaspersoft meets a broad range of reporting and BI requirements. In combination with our flexible licensing options, our application programming interfaces (APIs) differentiate Jaspersoft as the most powerful, flexible, and cost-effective reporting and BI solution for providers of corporate, open source, and commercial software applications.

This white paper introduces Jaspersoft's APIs to product managers and development managers. For an introduction to our editions and licensing options, please visit http://www.jaspersoft.com/editions.

# Built by Developers for Developers

Jaspersoft often describes its software as being "built by developers, for developers." Our namesake product, JasperReports, is not even an end-user application, but an open source reporting library that developers from all over the world have embedded in countless open source, commercial, and IT applications.

When we designed JasperServer, our JasperReports-based reporting and BI server for end-users, we stayed committed to the principles that made JasperReports so successful: applications should be built on reusable APIs that are easy to access, implement, and customize. As a result, we designed our APIs first, and then built JasperServer on top of them. You can work confidently with the Jaspersoft APIs, because our own applications are genuine reference implementations.

The Jaspersoft APIs are essential to the success of our business. We hope that this white paper will inspire you to think about how you, too, can use the Jaspersoft APIs to:

- **Embed** world-class reporting and analytics in your applications
- **Customize and extend** JasperServer's presentation logic to meet your customers' needs, without affecting source code or compromising forward compatibility
- **Integrate** JasperServer with other business applications and processes
- Allow **third party** web sites, portals, blogs, etc., to embed your dashboards, reports, and analyses

# JasperReports

As mentioned above, Jaspersoft is named after JasperReports, the world's most widely used open source reporting library. Developers can add the JasperReports library (a JAR file) to an application's Java CLASSPATH. At this point, JasperReports is "embedded" in the application, as if the reporting component were written by the host application's development team. Developers can then create, display, and export reports using the published JasperReports APIs.

## API Summary

| | |
|---|---|
| API | • JasperReports |
| Target users | • Java developers |
| Host application technologies | • Java servlet (for web applications)<br>• Java standalone (for desktop applications)<br>• Java command line (useful for testing) |
| Purpose | • Generate reports for screen and print |
| API documentation | • Download and unzip the latest jasperreports-x.x.x-project.zip file. API documentation is available under \jasperreports-x.x.x\dist\javadoc. |

# JasperServer APIs

JasperServer[1] is a comprehensive reporting and BI server. Its architecture includes a database repository for securing reports, analysis views, and associated resources; web user interfaces for doing things like ad hoc report design and repository management; pluggable back-end implementation engines to perform specific jobs, such as generating reports and authenticating users; and APIs that are used by business applications to access the underlying BI engines.

[1] Note: JasperReports is embedded in JasperServer. JasperServer calls JasperReports APIs to generate, display, and export reports.

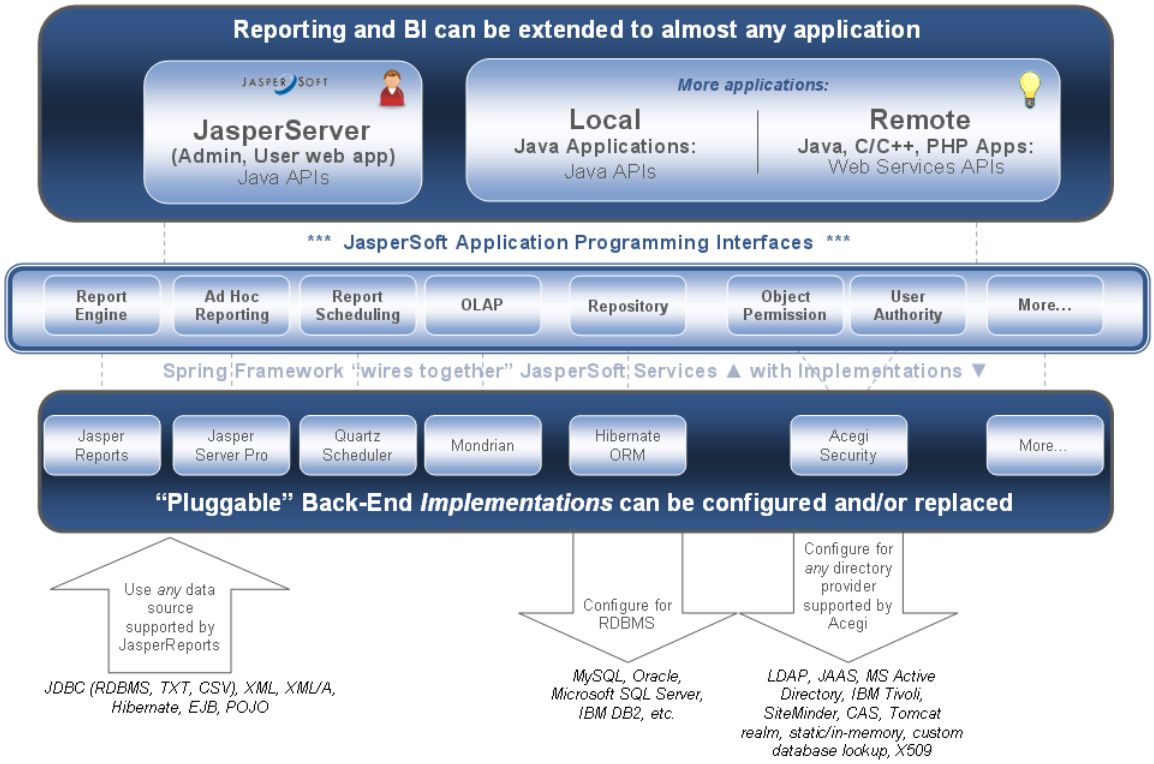


**Figure 1: All JasperServer functionality is exposed through APIs.**

Notice in Figure 1 that there are two categories for the front-end presentation layer: "Jaspersoft," represented here by JasperServer, and "More applications." What this means is that you can use the exact same APIs that Jaspersoft uses to develop your own applications.

## Embed Remote Report Management: Web Services API

JasperServer's managed reporting functions are exposed through web services APIs. Use the web services APIs when:

- Your business application is written in **C/C++, C#, PHP, Perl, Python**, **Ruby on Rails,** or any language other than Java (thus requiring a common communication protocol between your application and JasperServer).

  or

- Your business application runs **remotely**, even if it is written in **Java** (on a different computer or network than the JasperServer, thus requiring http to communicate between the two systems).

Using Jaspersoft's web services APIs, you can seamlessly integrate the following Jaspersoft-enabled functionality into your own applications:

**Report execution and repository management:**

| | |
|---|---|
| • **runReport** | Executes a report on the server, and returns the report's results in the specified format. The client application is responsible for prompting users for values to pass to any input controls defined in the report. |
| • **put** | Adds new resources to the repository or modifies existing ones. |
| • **get** | Obtains information about a resource. In the case of file resources, such as images, fonts, JRXML files, and JAR files, the resource file is attached to the response message. |
| • **list** | Retrieve a list of resources in a specified folder or report unit. |
| • **delete** | Deletes resources from the JasperServer repository. |

**Report scheduling:**

| | |
|---|---|
| • **deleteJob** | Delete a single report job specified by its ID. |
| • **deleteJobs** | Delete multiple report jobs specified by their IDs. |
| • **getJob** | Returns the full job details of a report job whose ID is sent as a parameter. |
| • **scheduleJob** | Schedules a new job. The job details must be sent as parameters; the operation returns the saved job details as its result. |
| • **updateJob** | Updates an existing job. The full job details (as retrieved using "getJob") must be sent as a parameter; the operation returns the updated job details as saved by the JasperServer scheduling service. |
| • **getAllJobs** | Returns the list of all accessible report jobs |
| • **getReportJobs** | Returns the list of all accessible report jobs for a specific report (whose URL is sent as a parameter) |

### *Reference Implementations*

The iReport plug-in interfaces with JasperServer using web services APIs. Tasks such as getting a list of resources from the repository, retrieving a JRXML file (so that it can be modified using the iReport graphical report designer), and running report units are executed in iReport using web services calls to a remote JasperServer.

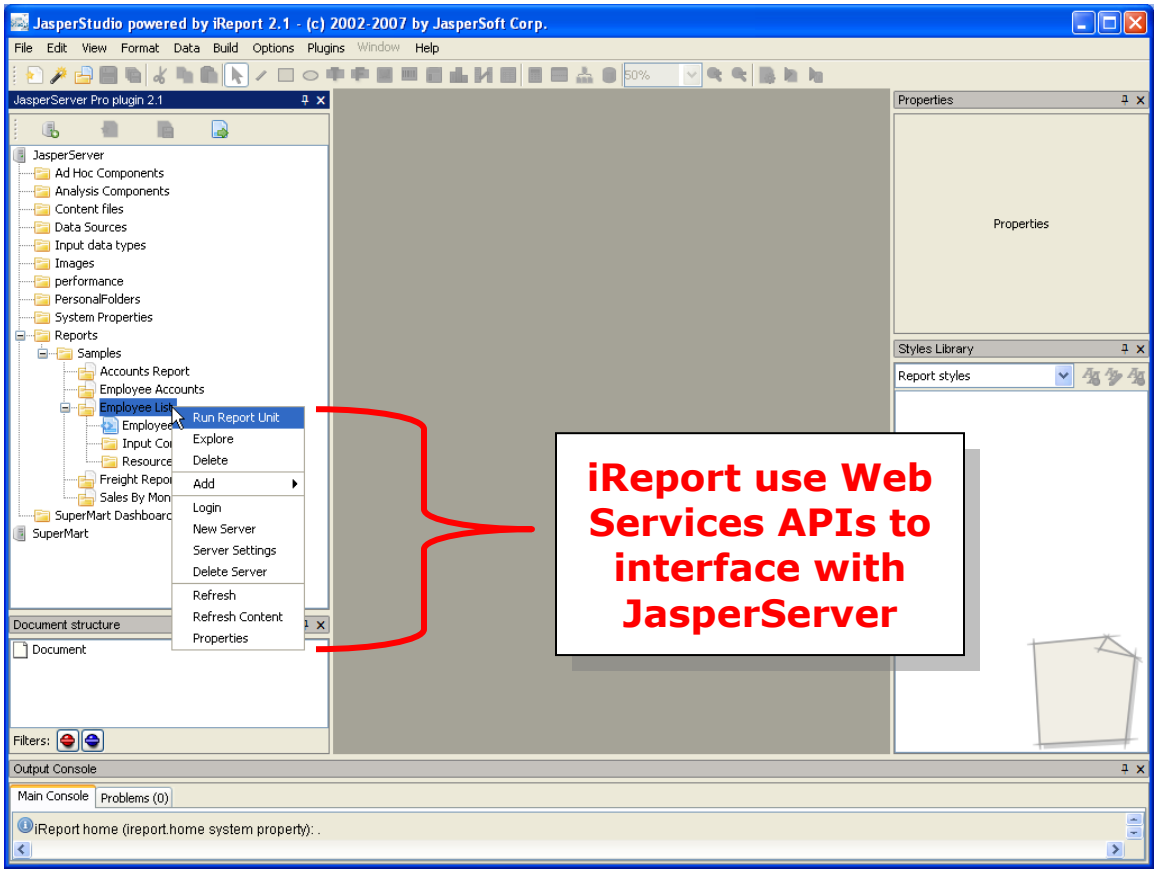


**Figure 2: iReport interface with JasperServer using a web services plugin.**

You can learn more about the iReport plugin by reading the *JasperServer Web Services Guide*. The guide is available online and is installed with JasperServer as /jasperserver/docs/JasperServer-Web-Services-Guide.pdf.

Jaspersoft also provides Java, PHP, and C/C++ sample clients that implement the above functionality. If you use Java, PHP, or C/C++, you can use these samples as-is. For other languages, you can review these implementations as guidelines. Samples are installed in /jasperver/samples.

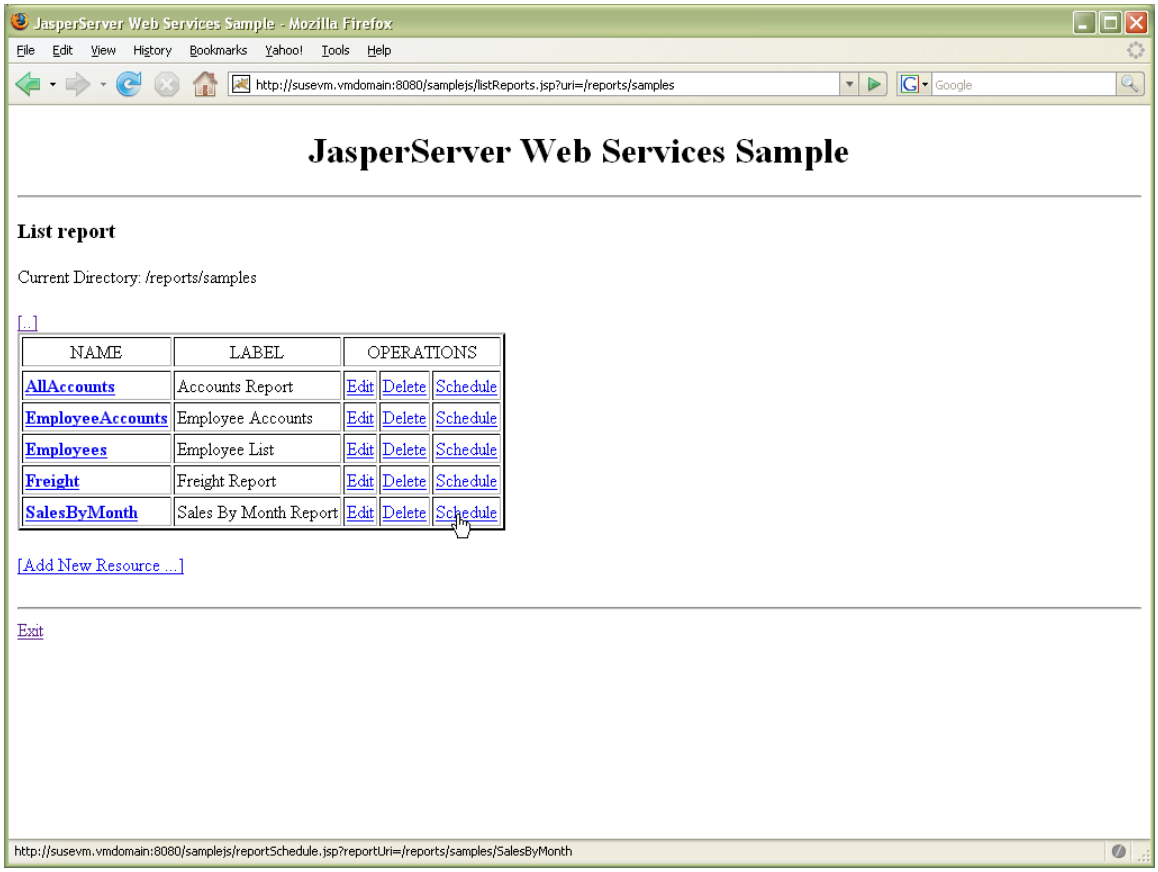


**Figure 3: Scheduling a report using a web services sample application.**

## *API Summary*

| | |
|---|---|
| API | • JasperServer web services |
| Target users | • Any developer |
| Host application technologies | • Any application that can make a web services call |
| Primary Uses | • Embed managed reporting in applications<br>• Embed report scheduling and distribution in applications |
| API documentation | • *JasperServer Web Services Guide*, available with the software as /jasperserver/docs/JasperServer-Web-Services-Guide.pdf. |

## The Complete API: Java

The complete JasperServer APIs are expressed as abstract Java interfaces that are connected to concrete implementations at run time using the Spring framework. The Java APIs are appropriate when:

- Your application needs to **closely integrate** with JasperServer components (so it can access the JasperServer libraries through the local Java **CLASSPATH**).

  or

- You want to change or extend **underlying JasperServer functionality**.

Using the Java APIs, you have full access to all JasperServer functionality, including:

1. **Repository Service API:** Store, lookup, and retrieve content using the Jaspersoft repository. (Also available with web services APIs.)
2. **Report Execution Engine Service API:** Generate reports on-demand. (Also available with web services APIs.)
3. **Custom Data Source API**: Use data sources for reports that are not supplied by JasperServer.
4. **Report Scheduling Service API:** Execute, store, and distribute reports on a schedule. (Also available with web services APIs.)
5. **Ad Hoc Service API:** Create ad hoc tables, charts, and crosstabs using a web browser. (Jaspersoft Professional & Enterprise Edition only.)
6. **OLAP Service API:** OLAP server and model runtime management. (JasperServer with JasperAnalysis only.)
7. **User Authority Service API**: User and role management.
8. **Object Permission Service API:** Search, retrieve, and modify repository object permissions.

The Java APIs can be used to implement functionality that does not exist in JasperServer by providing alternate implementations of the base APIs and extending the underlying Spring configurations.

### *Reference Implementations*

**JasperServer community release** and **JasperServer Professional** are both built using the JasperServer Java APIs.

### *API Summary*

| | |
|---|---|
| API | • JasperServer Java APIs |
| Target users | • Java developers |
| Host application technologies | • Java servlet (for web applications)<br>• Java standalone (for desktop applications)<br>• Java command line |
| Purpose | • Embed managed reporting in applications<br>• Embed report scheduling and distribution in applications<br>• Embed ad hoc reporting in applications<br>• Embed OLAP / analytics to applications<br>• Embed JasperServer repository administrative capabilities in an application |
| API documentation | • JavaDoc<br>• JasperServer Ultimate Guide[2]<br>• JasperServer Professional Administration Guide |

## Custom Ad Hoc Launcher

The Ad Hoc Launcher API for JasperServer Professional allows you to present a custom entry point into the ad hoc editor. This is a very useful API for applications that wish to provide end-users with ad hoc reporting capabilities. Some examples of possible Ad Hoc launchers include the following:

- A generic SQL query builder that allows power users to choose a JDBC data source, view the tables available from it, and **build a query** against one or more tables. This approach is demonstrated in the sample Ad Hoc launcher called browseDB, described below.

[2] The JasperServer Ultimate Guide is included in JasperServer Professional annual subscription fees. Open Source users can purchase the JasperServer Ultimate Guide from http://www.jaspersoft.com/jasperstore.

- An interface for **passing queries** based on application-specific metadata, such as a list of predefined queries maintained in its own table.
- An interface for constructing queries against a **custom datasource** which has metadata facilities, such as Hibernate.
- A **blank query box** in which a power user manually enters query text and report field definitions. This would be analogous to creating a JasperReport by editing the JRXML.

A custom Ad Hoc launcher must create an instance of the AdhocData Java class, which includes data source, query, and field information. This is passed to Ad Hoc through the servlet session. The user interface presented for building the AdhocData is completely up to you. For more information, refer to the *JasperServer Ultimate Guide*.

### *Proof of Concept*

JasperServer includes a proof-of-concept extension to JasperServer Professional that implements the Ad Hoc Launcher API. The browseDB sample application is a stripped-down ad hoc query builder.

**Example 1: Snippet of the BrowseDB controller source file, showing the Ad Hoc Launcher API.**

```
ModelAndView mv = new ModelAndView("redirect:/flow.html?
_flowId=adhocFlow&adhocLauncher=true");
```

You can build and deploy browseDB from the source files located in the JasperServer installation directory under /samples/customAdHoc.

**Example 2: Running browseDB, the proof-of-concept extension for the Ad Hoc Launcher API.**

```
http://server:8080/jasperserver-pro/browseDB/browseDB.html?
action=pickDatasource
```

### *API Summary*

| | |
|---|---|
| API | • Custom Ad Hoc Launcher |
| Target users | • Java developers |
| Host application technologies | • Java servlet (for web applications) |
| Purpose | • Provide a custom entry-point to the ad hoc editor |

| | |
|---|---|
| API documentation | • JavaDoc<br>• JasperServer Ultimate Guide[3]<br>• /samples/customAdHoc |

## HTTP API for the Ad Hoc Editor

**JasperServer Professional** includes an HTTP API that developers can use to modify the Ad Hoc Editor or create your own custom Ad Hoc Editor. The JasperServer Professional default Ad Hoc Editor is written in AJAX (JavaScript and XML).

If you wish to create your own Ad Hoc Editor (for example, not use drag-and-drop, or to create an editor in a programming language other than Java), you could re-use the JasperServer state object on the server and make use of the dynamic content that is retrieved.

### *Reference Implementation*

**JasperServer Professional** uses the HTTP API for the Ad Hoc Editor. The following are just a few examples of XMLHttpRequests that are made when working with the JasperServer Professional Ad Hoc Editor.

**Example 3: Adding a column to an ad hoc report of type "crosstab."**

```
http://server:8080/jasperserver-pro/adhoc/crosstab.html?
action=insertColumnGroup&f=ShipCountry&i-0
```

**Example 4: Changing the design theme for an ad hoc report of type "chart."**

```
http://server:8080/jasperserver-pro/adhoc/chart.html?
action=setThem&t=corporate
```

**Example 5: Clicking the "Run" button in the Ad Hoc Editor for reports of any type.**

```
http://server:8080/jasperserver-pro/adhoc/table.html
?action=saveTemp
```

[3] The JasperServer Ultimate Guide is included in JasperServer Professional annual subscription fees. Open Source users can purchase the JasperServer Ultimate Guide from http://www.jaspersoft.com/JasperShop_Documentation.html.

### *API Summary*

| | |
|---|---|
| API | • HTTP API for JasperServer Professional Ad Hoc Editor |
| Target users | • GUI developers |
| Host application technologies | • JasperServer Professional (to customize/redesign the JasperServer Pro UI)<br>  ○ XMLHTTP (AJAX/JavaScript)<br>  ○ Java Servlet<br>  ○ Spring MVC<br>  ○ Spring Web Flow<br>• Any other programming language (to create a new UI) |
| Purpose | • Customize or redesign the user interface for the JasperServer Professional ad hoc report editor<br>• Create an ad hoc report editor in a new programming language |
| API documentation | • JasperServer Ultimate Guide |

## HTTP Interface

The HTTP interface is the most commonly used API, and also the easiest to implement. However, the HTTP interface is not considered "embeddable" in the same way that the web services and Java APIs can be embedded in non-Jaspersoft applications. Rather, the HTTP interface is used primarily as "short-cuts" or entry points to commonly used JasperServer features or content.

For example, the SuperMart demo that ships with JasperServer Professional makes frequent use of the HTTP interface. It is this simple API that allows users to interact with the SuperMart dashboard, dynamically pass parameters from a JasperServer report to an OLAP view, display the user's personal folder based on the login ID, and more. The HTTP interface is also useful when emailing notification that scheduled jobs have completed.

As you'll notice in the examples that follow, the major APIs are:

- flow.html
- olap/**
- fileview/**

### *Link to Content*

The HTTP Interface can be used to link to JasperReports that were previously executed, exported, and saved to the JasperServer repository as PDF, HTML, Excel, or RTF content files.

**Example 6: Link to a PDF file stored in the JasperServer repository.**

```
http://server:8080/jasperserver-pro/fileview/fileview/
supermart/Details/CustomerDetail.pdf
```

### *View Resources in the JasperServer Repository*

**Example 7: Display all resources saved in a user's personal folder in the JasperServer repository.**

```
http://server:8080/jasperserver-pro/flow.html?_flowId=
repositoryFlow&folder=/PersonalFolders/demo
```

**Example 8: Display all resources of type "OLAP view" saved in all folders in the JasperServer repository.**

```
http://server:8080/jasperserver-pro/flow.html?_flowId=
olapViewListFlow
```

### *Execute JasperReports ReportUnits*

Execute and export JasperReports using the JasperServer web application.

**Example 9: Execute a JasperReports report without any parameters.**

```
http://server:8080/jasperserver-pro/flow.html?_flowId=
viewReportFlow&reportUnit=/supermart/Details/CustomerDashbo
ard
```

**Example 10: Execute the same report, but pass "4012" as an input control parameter and export to PDF instead of HTML.**

```
http://server:8080/jasperserver-pro/flow.html?_flowId=
viewReportFlow&reportUnit=/supermart/Details/CustomerDashbo
ard&customerID=4012&output=pdf
```

### *Execute JasperAnalysis OLAP Views*

Open, export, and save JasperAnalysis OLAP views using the JasperServer web application.

**Example 11: Open an OLAP view.**

```
http://server:8080/jasperserver-pro/olap/viewOlap.html?name=/supermart/RevenueAndProfit/Revenue_by_Store_Type&new=true
```

Note: You can also pass a new query to the OLAP view using the "&mdx=<query>" parameter.

### *Initiate the JasperServer Professional Ad Hoc Editor*

**Example 12: Display the ad hoc topics page.**

```
http://server:8080/jasperserver-pro/flow.html?_flowId=adhocFlow
```

**Example 13: Retrieve an ad hoc report unit from the repository and open it in the Ad Hoc Editor.**

```
http://server:8080/jasperserver-pro/flow.html?_flowId=adhocFlow&adhocReport=/adhoc/aru/Adhoc_Report
```

## About Security

It is important to keep in mind that the JasperServer repository is a secure repository, based on user profiles and roles. The default JasperServer installation requires users to authenticate; if they link to or try to update a resource for which they are not authorized, JasperServer returns an "access denied" message. Security is applied consistently across all access paths: browser, web services, and AJAX (JavaScript/XML). Security is implemented using Spring Acegi, and is therefore highly configurable to allow the use of non-standard and custom authentication and authorization services, single sign on, restricted web page access, and more. For more information on security configuration, including anonymous access, contact Jaspersoft.

# Jaspersoft API Benefits

The Jaspersoft APIs offer the fastest and most cost-effective way to incorporate professional, full-featured, and secure business intelligence into any application. With our APIs, providers of corporate, open source, and commercial software applications can:

- **Enter new markets with a differentiated product** that can include managed reporting, ad hoc reporting, and high-value analytics.
- **Accelerate delivery schedules** by customizing a complete solution rather than building and maintaining BI functionality in-house.
- **Build a competitive advantage and improve the value of your applications** by seamlessly integrating high quality reporting and BI components.
- **Reduce development, support, and overall COGS to improve profitability** by embedding a fully supported, globally proven reporting and BI solution.

To learn more about the Jaspersoft APIs, try the examples and review the API Summary sections in this white paper. Developers and product managers interested in embedding Jaspersoft technology into corporate, open source, and commercial applications can begin to learn about our flexible licensing options at http://www.jaspersoft.com/developers/.

Jaspersoft provides the most flexible, cost effective and widely deployed Business Intelligence suite in the world, enabling better decision making through highly interactive, web-based reports, dashboards and analysis. Leveraging a commercial open source business model, Jaspersoft provides end-to-end BI capabilities at a fraction of the cost of other vendors. The BI suite includes pixel-perfect enterprise reporting, ad hoc query, dashboards, OLAP and in-memory analysis, and data integration. Jaspersoft is the only BI vendor that enables companies to adapt to the new, virtualized world by providing a complete spectrum of on-premise, multi-tenant SaaS and cloud-based deployment options for both embedded and standalone business intelligence. Unlike traditional BI vendors, Jaspersoft is built on a modern, light-weight, standards-based architecture and offers greater vendor independence thanks to its open source codebase. Unlike niche BI vendors, Jaspersoft represents a safe choice with tens of thousands of production deployments across a wide range of industries.

Jaspersoft's open source business intelligence software has more than 11 million product downloads worldwide, 160,000 production deployments in 100 countries

and over 13,000 commercial customers. Its BI suite is advanced regularly by a development community of more than 150,000 registered members. For more information visit: www.jaspersoft.com and www.jasperforge.org.