

De03. Domino Web Dev Options - Which one to choose?

von Bernd Hort, 04.03.2020

Engage



Bernd Hort

- Master degree in computer science
- Lotus Notes developer since 3.3
- „It’s all about giving the user the right tool to get the job done.“

So many choices!

What is the right approach?





What is best for
my client?

Domino Web Dev Options - Which one to choose?

- Classic Domino Web
- XPages
- REST-API using Domino Access Services
- REST-API using Java
- HCL Domino AppDev Pack
- Project Keep

Stories matter

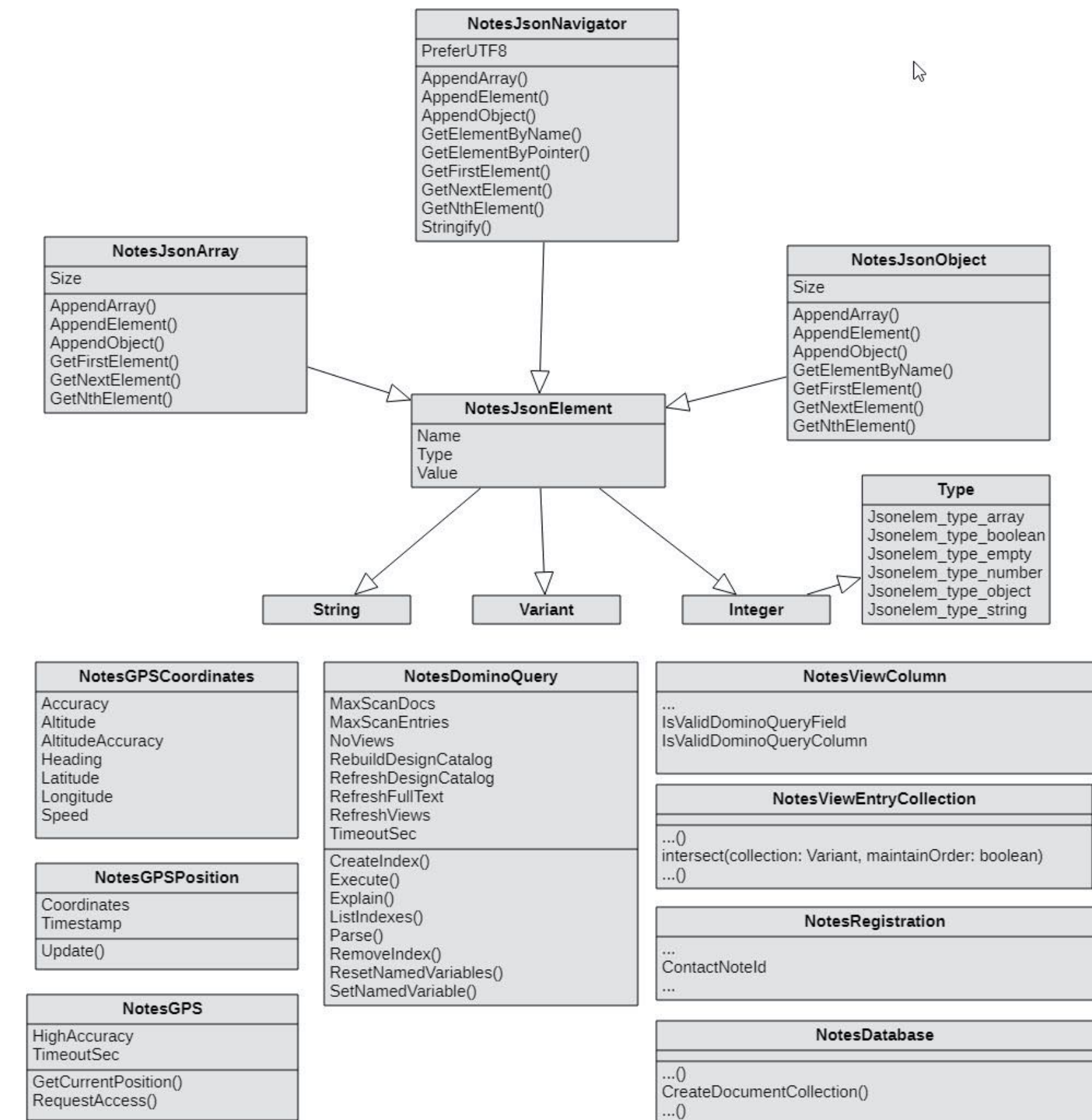


Photo by [Suzy Hazelwood](#) from [Pexels](#)

Classic Domino Web

Classic Domino Web

- Still relevant for
 - Small tasks
 - If you have lot of LotusScript Code
 - You are really good at LotusScript
- New LotusScript classes for Json handling



Classic Domino Web - Agents

- WebQueryOpen- and WebQuerySave-Agents for Forms
- With session.DocumentContext access to the current web form
- The items Query_String (GET) and Request_Content (POST) contains the raw request data
- The Agent Manager loads the Agent for every request

Pros and Cons



- Fast starting time
- Reuse of LotusScript



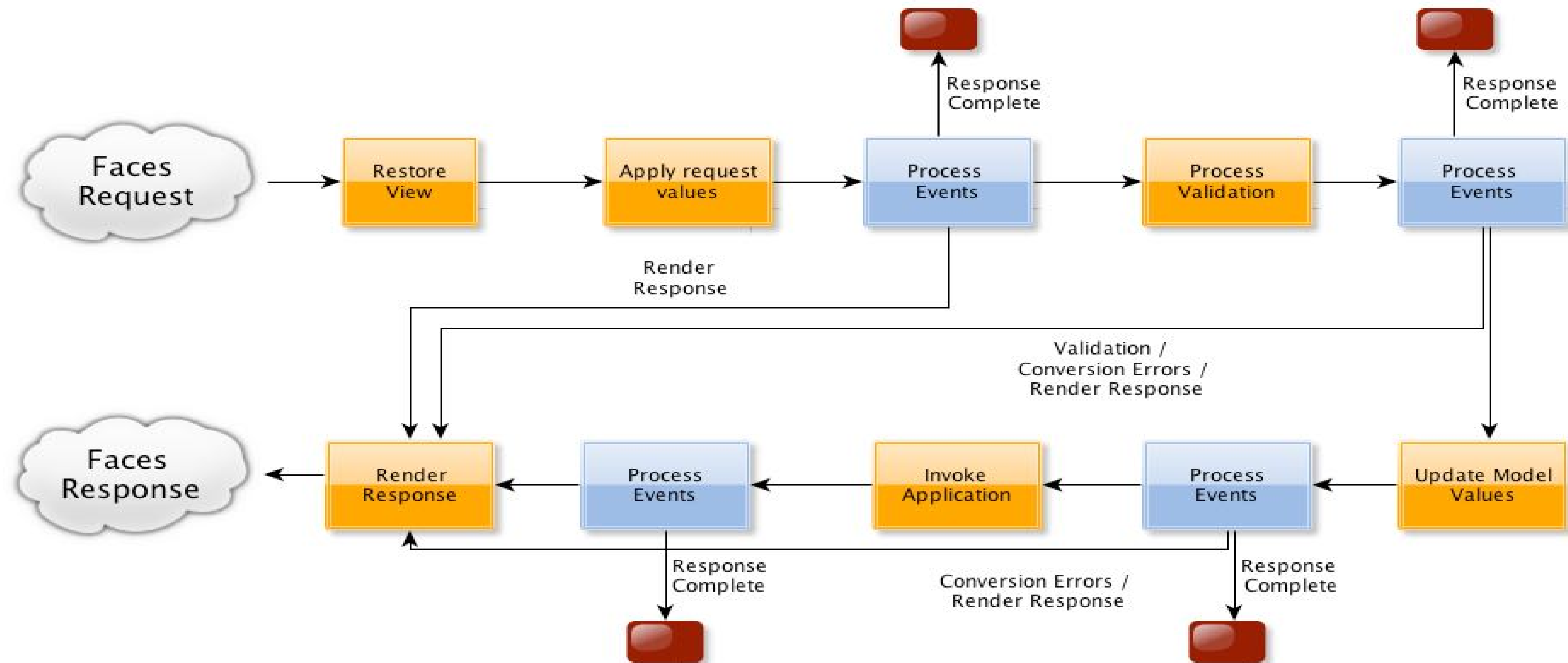
- No Caching
- Not suitable for large projects
- Not suitable for large teams

XPages



- Based on JavaServer Faces (JSF)
- Code base was JSF 1.1 with some features from later versions
- No new features for quite some time
- Every element on a web page has an object representation on the Java side

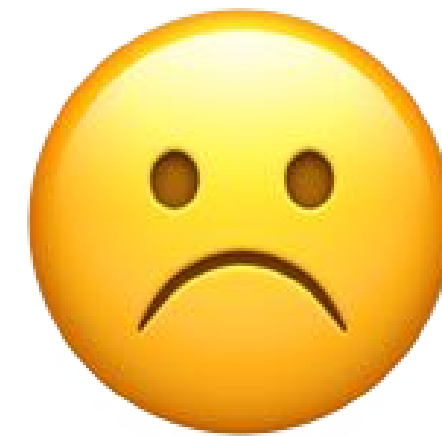
JSF Lifecycle



Pros and Cons



- Simplest way to get Domino data on a web page
- Drag & Drop support
- Good handling of MIME / Rich Text
- Close relationship between frontend and backend input validation



- Round trip to the server for every partial refresh
- Stateful
- No new features
- Community support gets less
- Plans of HCL unclear

**REST APIs
are the base of
every modern
web application!**



REST-API using Domino Access Services (DAS)

- Build in services
 - Domino Data Services
 - Domino Calendar Services
 - Domino Mail Services
- Domino Access Services must be enabled on the server document or internet site documents
- Must be enabled on database and view level

Pros and Cons



- No need to code anything



- Direct access to internal structure
- Limited to data in views or selected document
- Input validation based on @Formula language
- Input validation must be switched on via query string

REST-API using Extension Library REST-Service

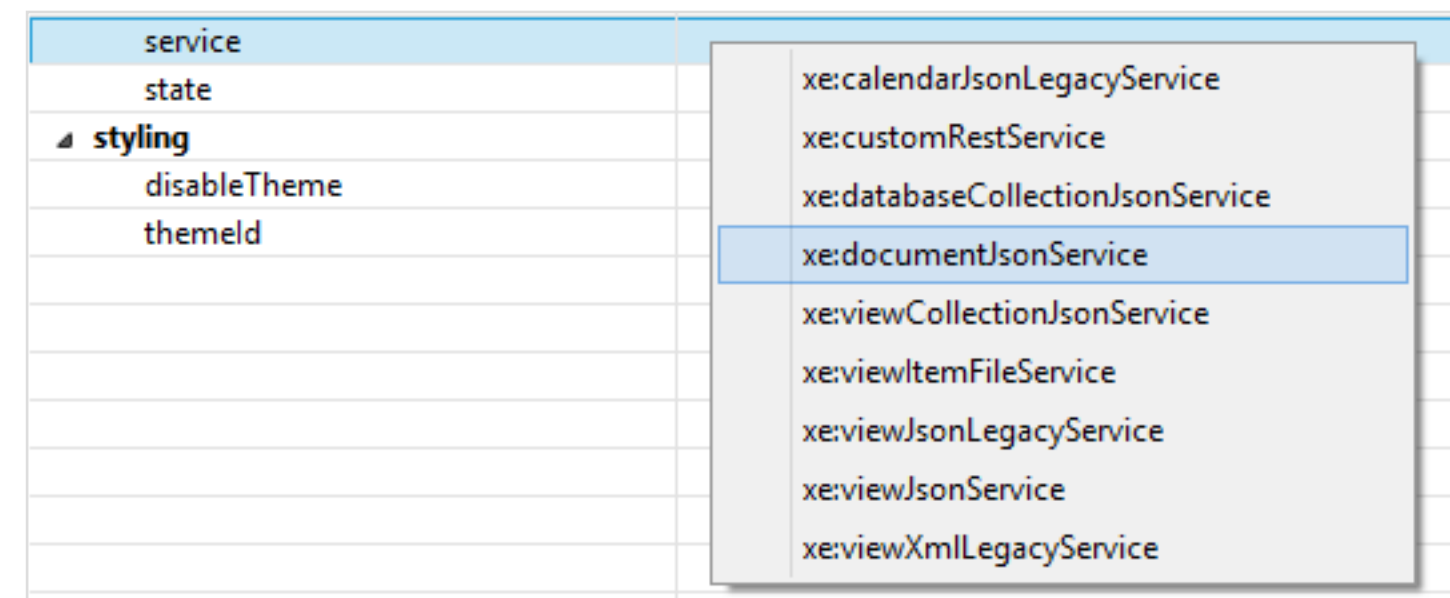
- REST-Service control as part of the Extension Library
- Must be part of a XPage
- Accessed via the path from the property pathInfo
 - {database}/{xpage-name}.xsp/{path}

The screenshot displays the SAP NetWeaver IDE interface. The main window shows the 'Properties' view for a 'REST Service' control. The 'All Properties' tab is active, displaying a table of properties and their values. The 'pathInfo' property is highlighted with a red box and contains the value 'customer'. The 'service' property is also highlighted and contains the value 'xe:documentJsonService'. On the right side, the 'Data Access' section of the 'Properties' view is expanded, showing 'JDBC Connection Manager' and 'Remote Service' options, with 'REST Service' selected and highlighted by a red box.

Property	Value
basics	
binding	
id	restService1
ignoreRequestParams	
jsId	
loaded	
pathInfo	customer
preventDojoStore	
rendered	
rendererType	
service	xe:documentJsonService
compact	
computeWithForm	true
contentType	application/json
databaseName	
defaultItems	true
documentUnid	
dojoAttributes	
dojoType	
formName	Customer
globalValues	
items	
loaded	
markRead	
parentId	
postDeleteDocument	
postNewDocument	
postOpenDocument	
postSaveDocument	

Extension Library REST-Service - Services

- calendarJsonLegacyService
- customRestService
- databaseCollectionJsonService
- documentJsonService
- viewItemFileService
- viewJsonLegacyService
- viewJsonService
- viewXmlLegacyService



Extension Library - xe:customRestService

- Full control over the REST service
- Direct implementation of methods
 - doDelete
 - doGet
 - doPost
 - doPut
- Or Custom Service Bean

Extension Library - Custom Service Bean

- The class needs to extend CustomServiceBean
- Access to everything a XPage Environment has
 - incl. SessionAsSigner

```
public class CustomerRESTServiceBean extends CustomServiceBean {  
    ...  
    @Override  
    public void renderService(CustomService service, RestServiceEngine engine)  
        throws ServiceException {  
  
        try {  
            HttpServletRequest request = engine.getHttpRequest();  
            HttpServletResponse response = engine.getHttpResponse();  
            response.setHeader("Content-Type",  
                "application/json; charset=UTF-8");  
  
            String method = request.getMethod();  
            if (method.equals("GET")) {  
                this.doGet(request, response);  
            } else if (method.equals("POST")) {  
                this.doPost(request, response);  
            } else if (method.equals("PUT")) {  
                this.doPut(request, response);  
            } else if (method.equals("DELETE")) {  
                this.doDelete(request, response);  
            }  
  
        } catch (Exception e) {  
            throw new RuntimeException(e);  
        }  
    }  
}
```

Pros and Cons



- Different levels of programming skills are supported
 - From build in services
 - To Java implementation
- With customRestService full control
- Business Logic
- Caching



- Bound to a XPage



Photo by Pixabay from Pexels

And then things got interesting!

Sven Hasselbach - XPages: Run your own Servlets

- XPages: Run your own Servlets
 - <http://hasselba.ch/blog/?p=746>
- Describes how to use a ServletFactory from the XPages runtime to create servlets
- Found on a Chinese IBM Developerworks blog entry
- <https://www.ibm.com/developerworks/cn/lotus/xpage-servlet/index.html>

新型 Web 2.0 应用：XPage 与传统 Servlet 的完美结合

在 XPage 中使用 Ajax

桂训龙
2010 年 3 月 04 日发布

分享 G+ 邮件 评论 2

概述

IBM Lotus Notes/Domino 8.5 为 Lotus Domino Web 2.0 应用程序开发人员提供了一种新的开发技术 — XPage，可以利用它创建 Web 2.0 应用。有关 XPage 的基础介绍，建议读者阅读文章“[XPage 全接触](#)”和教程“[在 Lotus Domino Designer 中利用 XPages 的威力](#)”。

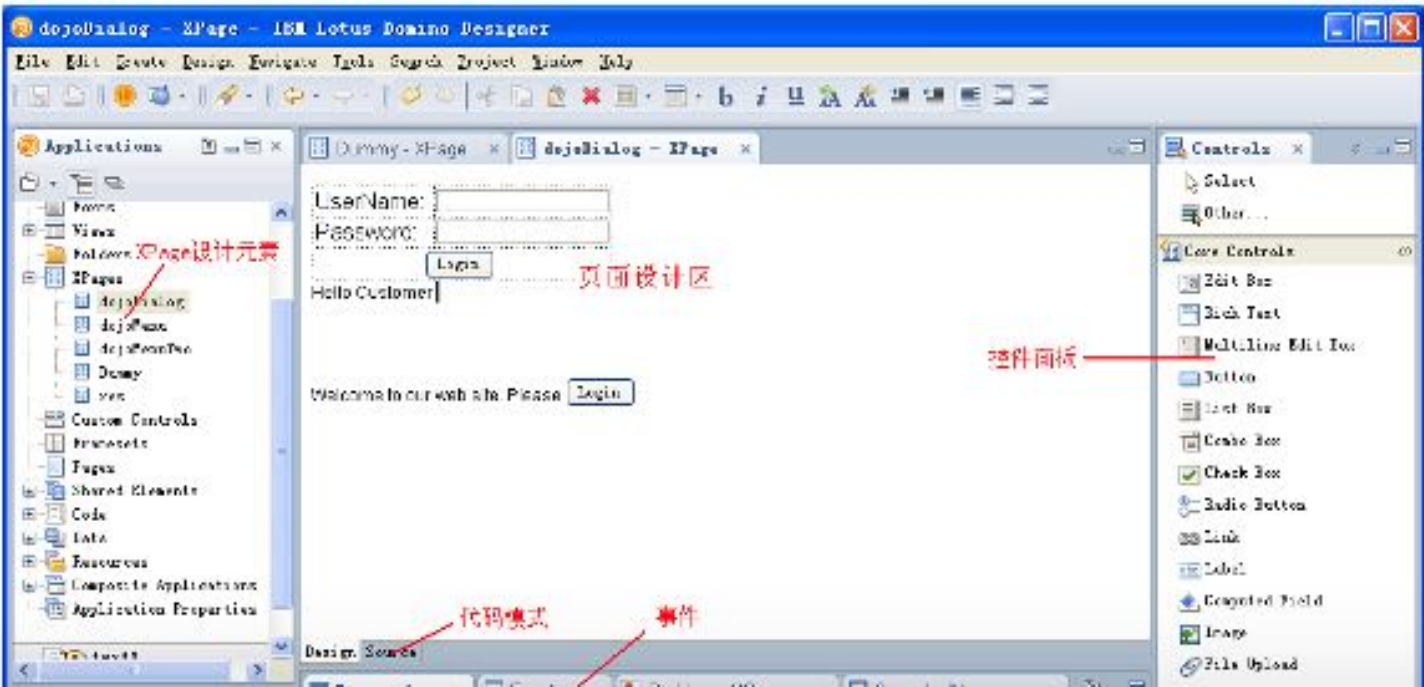
XPage 的前端采用 JSF（Java Server Face）技术实现，并封装了对 Dojo 控件的调用，与传统的 Lotus Domino Web 开发相比较，使用 XPage 开发的 Web 应用界面更加美观。当在 XPage 应用中创建基于 Ajax 的 Web 2.0 应用时，可以采用 XPage 控件及 Dojo 控件构建美观大方的前端界面。但是，还需要一个后端程序来处理业务逻辑，从而实现业务逻辑层及应用显示层的相分离。而 Servlet 在处理业务逻辑上功能强大，是个不错的选择。

接下来，本文将详细介绍如何使用 XPage 和 Servlet 技术来创建 Web 2.0 应用程序，我们将使用一个简单的用户登录应用场景来演示说明。

在 XPage 中使用 Dojo 创建 Ajax 客户端

Lotus Domino Designer 8.5.1 为用户提供了集成 Eclipse IDE 的可视化开发环境，有着强大的 Web 页面设计视图，程序设计人员可以利用它获得“所见即所得”的用户体验，如图 1 所示。

图 1.XPage 设计视图



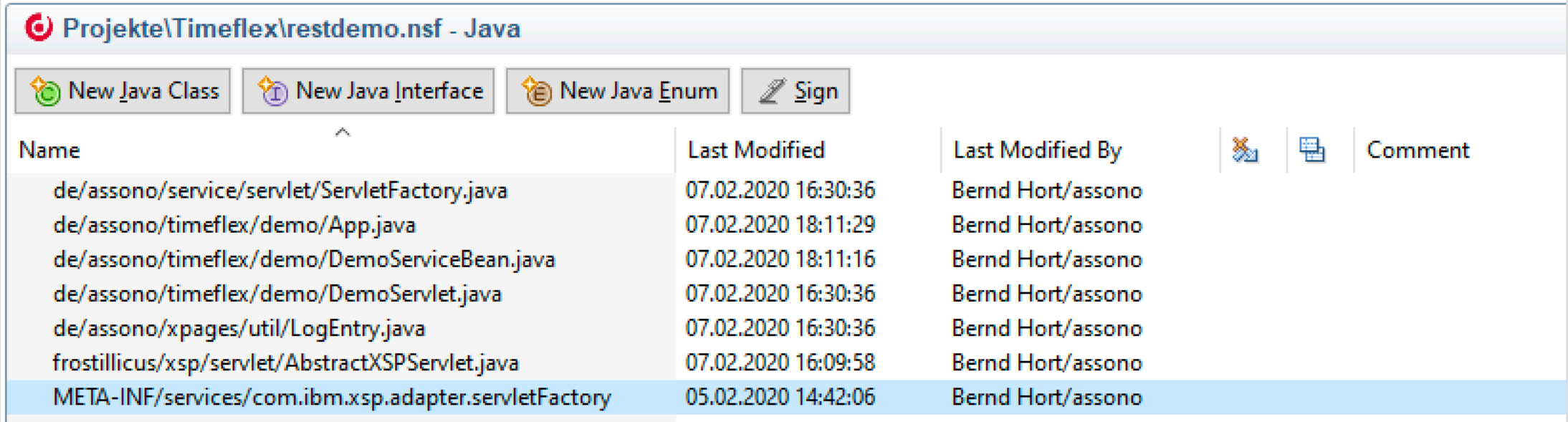
Lotus Domino Designer 8.5.1 免费软件下载

请立即下载免费使用的 Domino Web 2.0 开发工具：Lotus Domino Designer 8.5.1，尝试利用 XPage 结合 Servlet 技术开发第一个 Domino Web 2.0 应用程序。

- 免费软件下载：Lotus Domino Designer 8.5.1
- 试用版软件下载：Lotus Domino Server 8.5

The Servlet Factory

- The Servlet Factory is registered via a text file
 - The name has to be com.ibm.xsp.adapter.servletFactory
- Inside is the qualified Java class name of the Servlet Factory
- Must resist in a folder META-INF/services/
- The folder must be visible to the Java Build Path
- Extends the DefaultServletFactory
 - com.ibm.xsp.extlib.services.servlet.DefaultServletFactory



Name	Last Modified	Last Modified By	Comment
de/assono/service/servlet/ServletFactory.java	07.02.2020 16:30:36	Bernd Hort/assono	
de/assono/timeflex/demo/App.java	07.02.2020 18:11:29	Bernd Hort/assono	
de/assono/timeflex/demo/DemoServiceBean.java	07.02.2020 18:11:16	Bernd Hort/assono	
de/assono/timeflex/demo/DemoServlet.java	07.02.2020 16:30:36	Bernd Hort/assono	
de/assono/xpages/util/LogEntry.java	07.02.2020 16:30:36	Bernd Hort/assono	
frostillicus/xsp/servlet/AbstractXSPServlet.java	07.02.2020 16:09:58	Bernd Hort/assono	
META-INF/services/com.ibm.xsp.adapter.servletFactory	05.02.2020 14:42:06	Bernd Hort/assono	

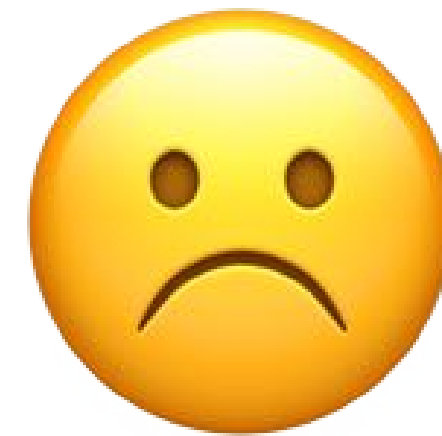
Jesse Gallagher - Building XPages servlets with FacesContext access

- Jesse Gallagher - Building XPages servlets with FacesContext access
 - <https://frostillic.us/blog/posts/159496067A27FD3585257A70005E7BC1>
- Based on Sven Hesselbachs blog and some work from Nathan Freeman and Niklas Heidloff described how to extended the DesignerFacesServlet class to get access to the FacesContext
- With the FacesContext access to the current Notes session and current database was possible
- Implemented an AbstractXSPServlet as entry point for own servlet implementations
- Not possible to use SessionAsSigner (due to some class loader problems)

Pros and Cons



- All Java features
- Full control
 - Business Logic
 - Caching
- Deployment based on Notes template



- Low level servlet implementation
- No SessionAsSigner

REST-API with Java (JAX-RS) using Jersey

- The Java API for RESTful Services (JAX-RS) is part of the Java spec
- The Sun / Oracles reference implementation Jersey contains all needed classes
 - Jersey version 1.71.1 is the last stable version that supports Java 6 and JAX-RS 1.1
- Extending the ServletContainer and Application from Jersey initialises the JAX-RS application



JAX-RS

- Dealing with higher abstract level
- Input parameters and return values are Java objects
- An object mapper converts Json objects in Java objects and vice versa
 - Jackson is a well known implementation



Java (JAX-RS) sample



```
@Path("/masterdata")
@Consumes("application/json")
@Produces("application/json; charset=UTF-8")
public class MasterDataController {
    @GET
    @Path("/countries")
    public Response getCountryList() {
        try {
            Response response;
            response = this.checkAnonymous();
            if (response != null) {
                return response;
            }

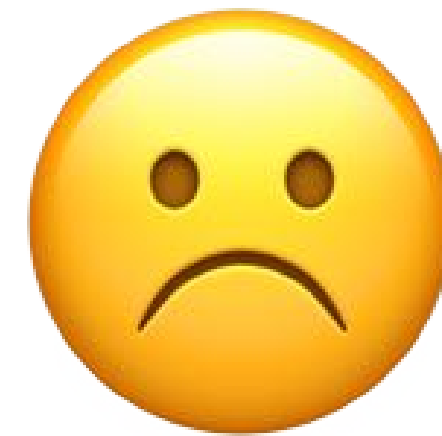
            MasterDataDAO masterDataDAO = Application.INSTANCE.getMasterDataDAO();

            return ResponseEnvelope.createSuccessResponse(masterDataDAO.getCountryList());
        } catch (Exception e) {
            return ResponseEnvelope.createErrorResponse(e.getMessage());
        }
    }
    [...]
}
```

Pros and Cons



- All Java features
- Full control
 - Business Logic
 - Caching
- Deployment based on Notes template
- Uses JAX-RS



- No SessionAsSigner

OSGi Plugins

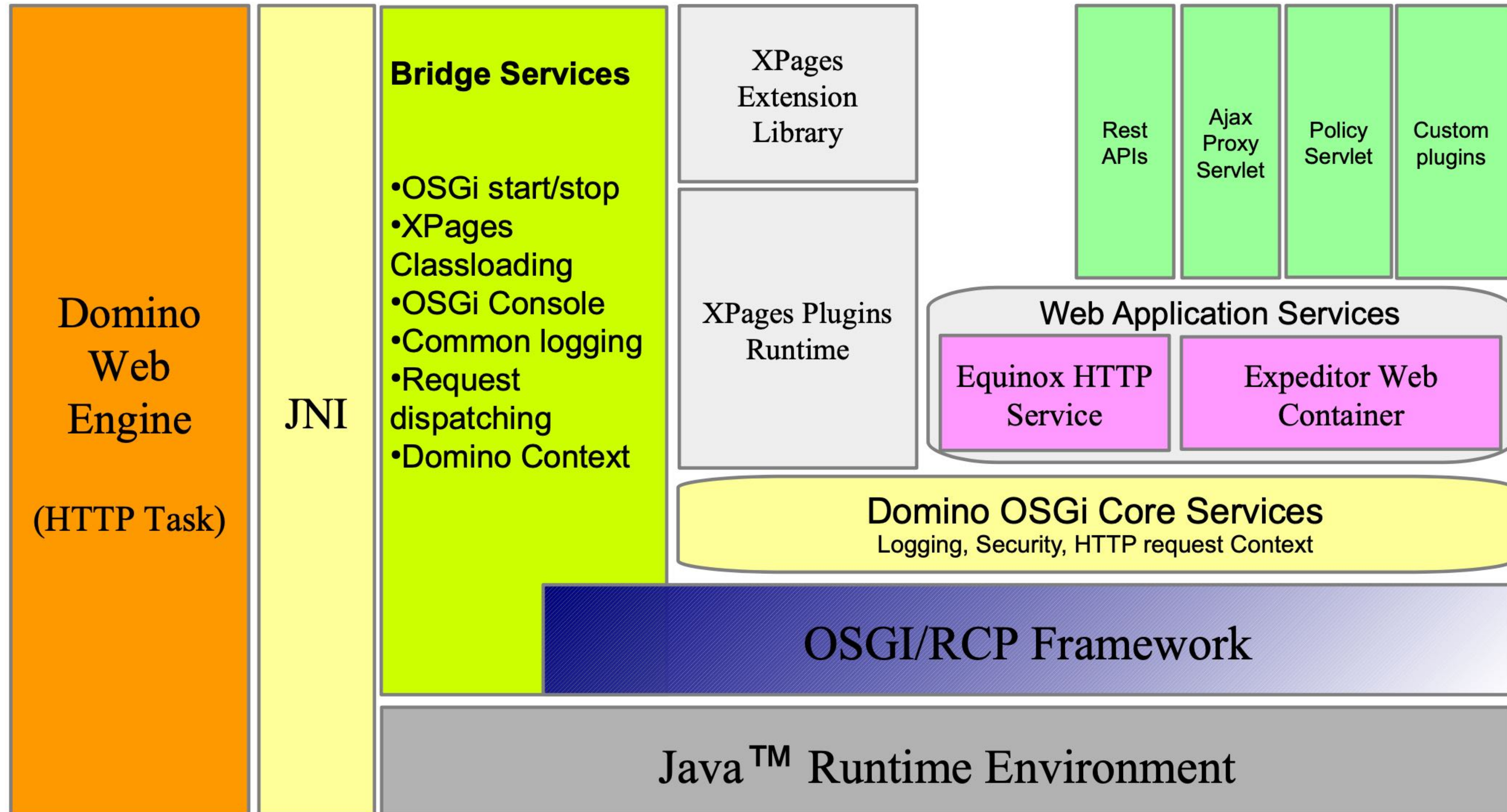


OSGi Plugins

- Since version 8.5.2 there is an OSGi integration in the HTTP task in Domino
- It is the base of the Extension Library, DOTS and many things more
- Paul Fiore (et al) had a session Lotusphere 2012
SHOW112 - Domino OSGi Development
 - <https://www.slideshare.net/fiorep/domino-osgi-development>



IBM Lotus Domino OSGi in HTTP architecture



Web Application Services

- There are two components which could be used for a REST API
- Equinox HTTP Service
 - More for simpler servlets
- Expeditor Web Container
 - Based on J2EE

OSGi - Multiple deployments

- File based
 - Bundles are packaged as jars or directories and dropped into the server file system
 - recommended location is {dominodata}/domino/workspace/applications
 - Bundles execute with maximum security privileges
- NSF based
 - Bundles are imported into a database using the UpdateSite.ntf template
 - Server is configured to use the updateSite database (notes.ini variable for HTTP)
 - Documents containing the bundles must be signed with privileged user id to load, after which bundles execute with maximum security privileges

Toby Samples

- Toby Samples replied to Sven Hasselbachs blog entry with =>
- Years later he wrote an article series „ JAX-RS or THE way to do REST in Domino“
- Using the Apache Wink implementation from the Extension Library to implement a JAX-RS servlet



Toby Samples sagt:

9. Juli 2012 um 16:02

This is definitely the easiest way to do so, but they only run in context of the db if I remember correctly. You can build real HTTPServlets that run in server context if you deploy them using an osgi bundle. Let me know if you want the info on how to do that.

[Antworten](#)



Sven Hasselbach sagt:

10. Juli 2012 um 08:57

Yes please! This would be really interesting, I have planned to integrate PHP into XPages. Can you send me an email with these infos to blog@hasselba.ch or post a link?

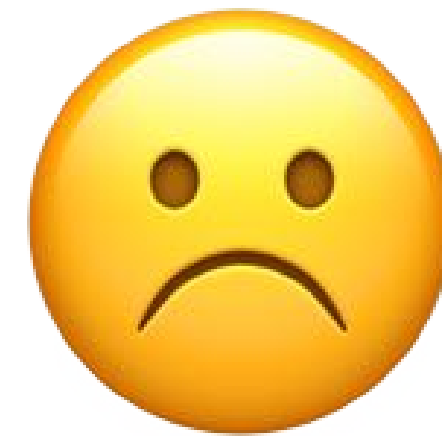
Thanks in advance!

[Antworten](#)

Pros and Cons



- All Java features
- Full control
 - Business Logic
 - Caching
- Uses JAX-RS
- Maximum security privileges



- Installing an OSGi plugin might be a problem
- Every update needs an HTTP task restart



A different
approach to Web
Apps:

vaadin}>

Vaadin}> - Build Progressive Web Apps in Java

- „Vaadin is an open source web framework that helps Java developers build great user experiences with minimal effort. Focus on creating apps that your users will love.“
- The complete programming happens in Java - backend and frontend



Sven Hasselbach - HowTo: Vaadin on Domino

- Sven Hasselbach also wrote an article „HowTo: Vaadin on Domino“
 - <http://hasselba.ch/blog/?p=1891>

Paul Withers - From XPages to Web App

- Paul Withers wrote an excellent article series „From XPages to Web App“
 - <https://www.intec.co.uk/from-xpages-to-web-app-introduction/>
- Covers all the way from XPages development, using OSGi and ending with using Vaadin to built a web app

Pros and Cons



- Writing frontend and backend code in Java.
- Installed as OSGi plugin
=> Maximum security privileges



- Installing an OSGi plugin might be a problem
- Every update needs an HTTP task restart
- Need to learn vaadin}>

Alternatives to Domino Java Implementation



OpenNTF Domino API

- Wrapper API for lotus.domino API
- Created by Jesse Gallagher, Nathan T Freeman, Paul S Withers
- <https://oda.openntf.org/>
- Installed as an OSGi Plugin (Domino Server and Domino Designer)
- Provided demo applications and demo OSGi plugins REST API
- Apache License 2.0



Domino JNA

- Cross-platform access to IBM/HCL Notes/Domino C API methods from Java
- From Karsten Lehmann / Mindoo GmbH
- On Github <https://github.com/klehmann/domino-jna>
- Installed as an OSGi Plugin (Domino Server and Domino Designer)
- Apache License 2.0



Business Logic outside of Domino

HCL Domino AppDev Pack

- The AppDev Pack adds Node.js support to HCL Domino Server.
 - <https://doc.cwpcollaboration.com/appdevpack>
- Three components
 - Node.js module **domino-db** - provide access to Domino data from Node.js
 - **Proton** Domino Task - listening to requests from domino-db
 - Node.js based service **IAM** (Identity and Access Management) for OAuth 2.0 Authentication

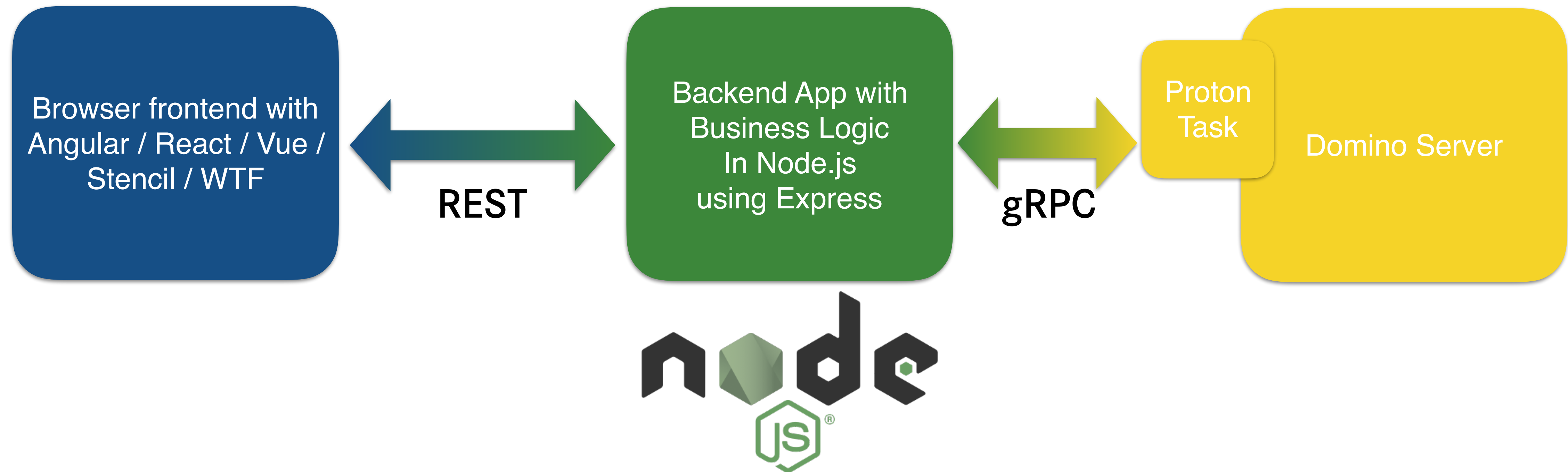


HCL Domino AppDev Pack

- From Node.js point of view Domino is just another NoSQL-Database
- domino-db is the database driver
- HCL solves with the Domino AppDev Pack two problems
 1. Attracts new developers
 2. Supports a modern HTTP server with HTTP2, WebSockets etc.
without the need to update the HTTP Server in Domino



HCL Domino AppDev Pack



Pros and Cons



- Using JavaScript for Backend implementation
- Express is a very powerful and performant HTTP Server
- Millions of Open Source projects, tutorials, tool etc.
- Still leveraging Domino security and data storage



- Domino only data storage

Project Keep - Secure access to Domino Design and Data

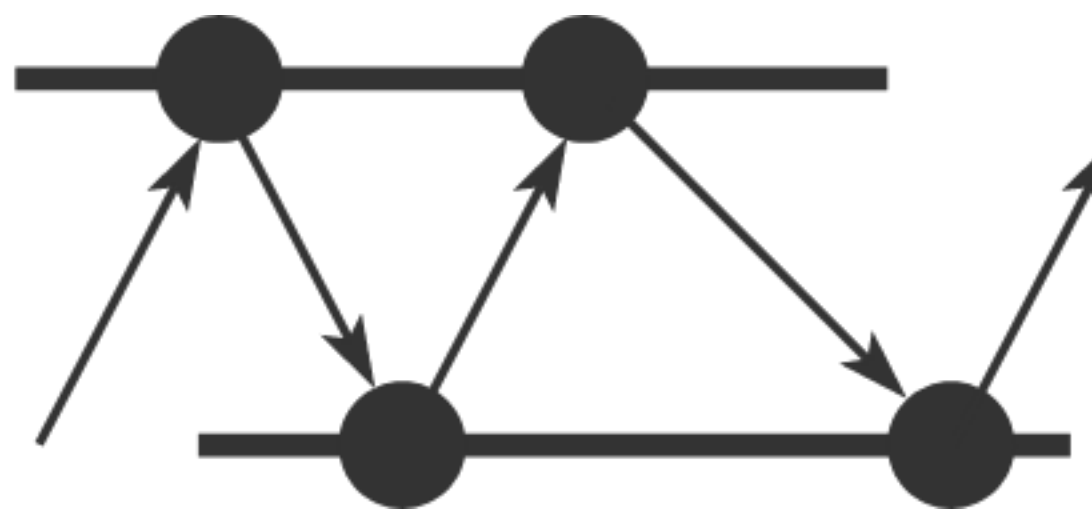
- REST-APIs for accessing Domino design and data
 - <https://frascati.projectkeep.io/>
- Very fast
- Based on Eclipse Vert.X
 - Brings it own HTTP server (incl. HTTP2)
- Uses Domino JNA
- Configure access to data down to field level
- Database ACL still defines the access level a user has
- Supports local users



Eclipse Vert.x



- Eclipse Vert.x is a tool-kit for building reactive applications on the JVM.
- Eclipse Vert.x is event driven and non blocking. This means your app can handle a lot of concurrency using a small number of kernel threads. Vert.x lets your app scale with minimal hardware.



Project Keep



Pros and Cons



- REST-API for every aspect of a Domino database
- Highly configurable
- Designed with security in mind
- Application server can be any server



- Domino only data storage



Questions?

Picture Credits

- Unplash
<https://unsplash.com/>
- Pexels
<https://www.pexels.com/>