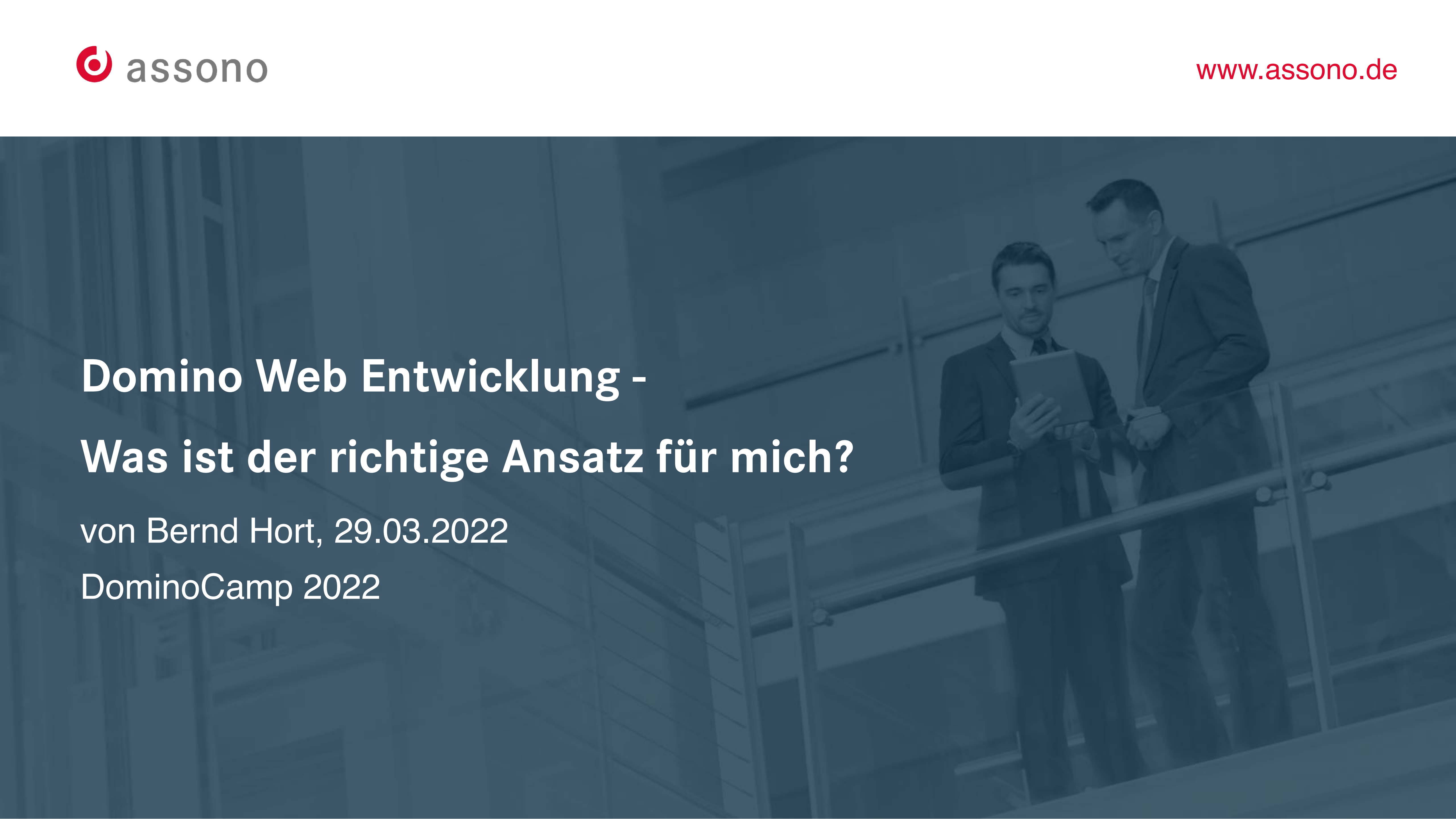


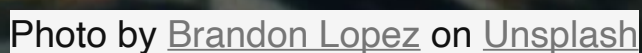


Domino Web Entwicklung - Was ist der richtige Ansatz für mich?

von Bernd Hort, 29.03.2022

DominoCamp 2022

Welches ist der richtige Ansatz?





Was ist das beste
für meinen
Kunden?

Domino Web Entwicklung - Was ist der beste Ansatz für mich?

- Nomad Web
- Klassische Domino Webentwicklung
- XPages
- REST-API mittels Domino Access Services
- REST-API mittels Java
- HCL Domino AppDev Pack
- HCL Domino REST-API aka. Project Keep





Bernd Hort

- Diplom Informatiker
- Notes Entwickler seit Version 3.3
- „Gib dem Anwender das richtige Werkzeug die Aufgabe zu erledigen“



Nomad Web

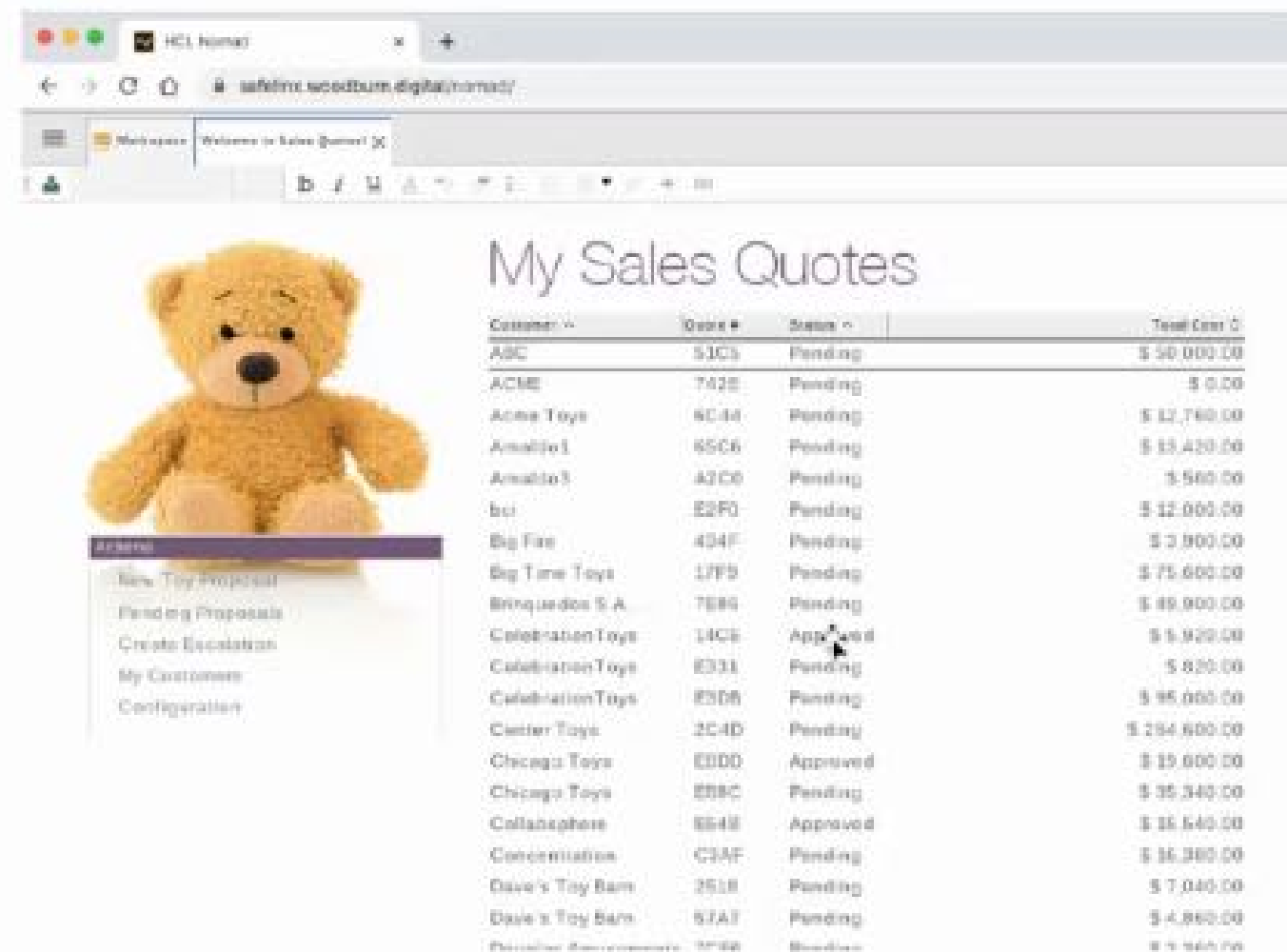
Photo by [Belinda Fewings](#) on [Unsplash](#)

Nomad Web

Nomad Web (Domino v12) Summary

HCL

- Customers want a browser-based solution to run their existing Apps
- It removes the need for their Citrix farms investments
- Reduce the deployment cost of deploying the Notes client
- Beta to be announced tomorrow...



Chrome



Firefox



Safari



Edge

Coming soon.



Windows

macOS



assono

Nomad Web

Nomad Web - Technology Overview

HCL

C / C++ Code

```
#ifndef NSF_DEFS
#define NSF_DEFS

/* Note Storage File Glob

#include "nsfdata.h"
#include "nsfdb.h"
#include "nsfnote.h"
#include "nsfsearc.h"
#include "nsferr.h"
```



emscripten

WebAssembly



WASM module

JavaScript



App HTML/PWA



Browser



Workspace X

Nomad Team Room - Team Docume... X

TeamRoom

New Document

New Response

New Response to Response

Database Help

Nomad

Leader/Facilitator Options

Team Documents

Personal Documents

Project Information

Date ^

Topic

Due ↗

Author ^

▼ All

06/09/2020 Merging fixes from Nomad to Master Jeremy Figgins/USA/PNPHCL

06/09/2020 Code Branch scheme Rick Gillasp/USA/PNPHCL

05/27/2020 Use cases for reauth in Nomad client Doug Conmy/USA/PNPHCL

03/20/2020 Nomad Initial configuration flow notes Rick Gillasp/USA/PNPHCL

11/22/2019 Nomad compilation and checks vs Stand Doug Conmy/USA/PNPHCL

10/31/2019 Gaps between C code and Eclipse for Ai Doug Conmy/USA/PNPHCL

10/23/2019 Collabsphere 2019 Deck Maxx Sutton/USA/PNPHCL

07/10/2019 Nomad and VPN solutions 07/10/2019 Doug Conmy/USA/PNPHCL

05/28/2019 Notes/Domino Source Directory and Dev Robert S Sielken/USA/PNPHCL

▼ Android

04/08/2020 Recorded Narrated Mobile Demo Videos Maxx Sutton/USA/PNPHCL

03/19/2020 Android WSS dark enable instructions Phil Loats/USA/PNPHCL

02/24/2020 Device resolution and specs Phil Loats/USA/PNPHCL

▼ 02/17/2020 3 Nomad Mobile WSS requirements and dei Phil Loats/USA/PNPHCL



Copyright © 2020 HCL Technologies Limited | www.hcltechsw.com

HCL SOFTWARE

assono

Pro und Kontra



- Bestehende Anwendungen laufen einfach
- Wiederverwendung von LotusScript



- Bestehende Anwendungen laufen einfach
- Keine Java-Agents
- Kein OLE/COM
- Keine MS Office Integration



Stories matter

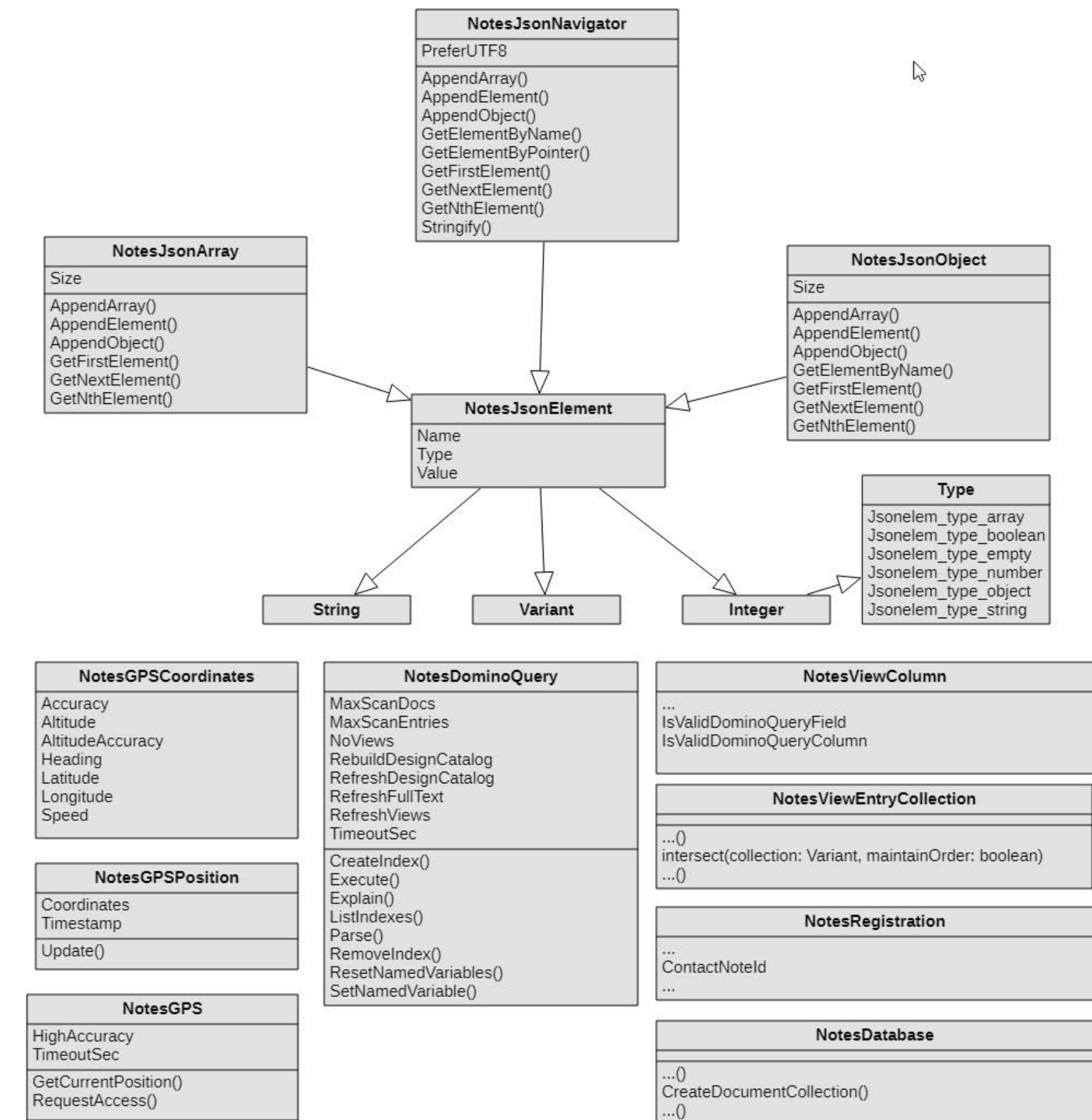


Photo by [Suzy Hazelwood](#) from [Pexels](#)

Klassische Domino Webentwicklung

Klassische Domino Webentwicklung

- Immer noch relevant für
 - Kleine Aufgabenstellungen
 - Große LotusScript Code Basis
 - Sehr gute LotusScript-Kenntnisse
- Neue LotusScript Klassen für Json Handling



Klassische Domino Webentwicklung - Agenten

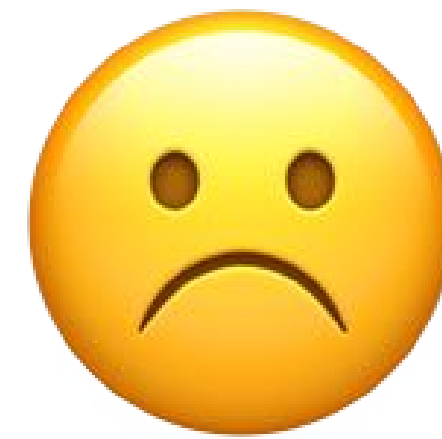
- WebQueryOpen- und WebQuerySave-Agenten für Masken
- Mit session.DocumentContext Zugriff auf das aktuelle Dokument
- Die Items Query_String (GET) und Request_Content (POST) enthalten die „raw“ Daten
- Der Agent Manager lädt den Agent für jede Anfrage



Pro und Kontra



- Sehr schnelle Startzeit
- Wiederverwendung von LotusScript



- Kein Caching
- Nicht geeignet für große Projekte
- Nicht geeignet für große Teams

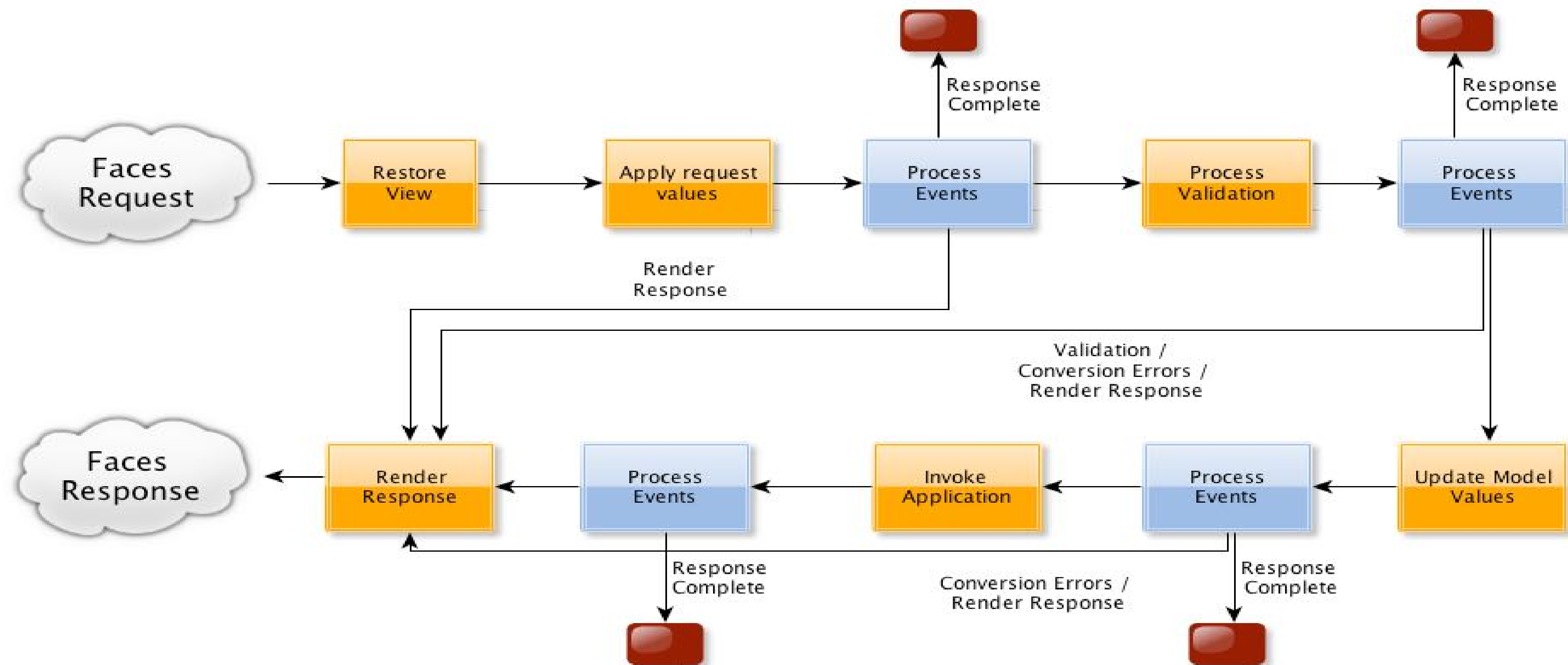




- Basierend auf JavaServer Faces (JSF)
- Die Code bases war JSF 1.1 mit einigen Features von späteren Versionen
- Seit einiger Zeit keine neue Features
 - Ausnahme ist die Unterstützung für Bootstrap 4 in Version 12
- Jedes Element auf der Webseite hat eine Objekt-Representation auf der Java Seite



JSF Lifecycle



Pro und Kontra



- Einfachster Weg um Domino Daten auf eine Webseite zu bringen
- Drag & Drop Support
- Gutes Handling von MIME / Rich Text
- Enge Beziehung zwischen Frontend und Backend Eingabevalidierung



- Round-trip zum Server für jeden „Partial-Refresh“
- Stateful
- Keine neuen Features
- Community Support wird weniger
- Pläne von HCL sind unklar



**REST APIs
sind die Basis von
modernen Web-
Applications!**



REST-API mittels Domino Access Services (DAS)

- Integrierte Services
 - Domino Data Services
 - Domino Calendar Services
 - Domino Mail Services
- Domino Access Services muss im Server-Dokument oder im Internet-Site-Dokument aktiviert werden
- Muss auf Datenbank- und Ansichtsebene aktiviert werden



Pro und Kontra



- Keine Programmierung notwendig

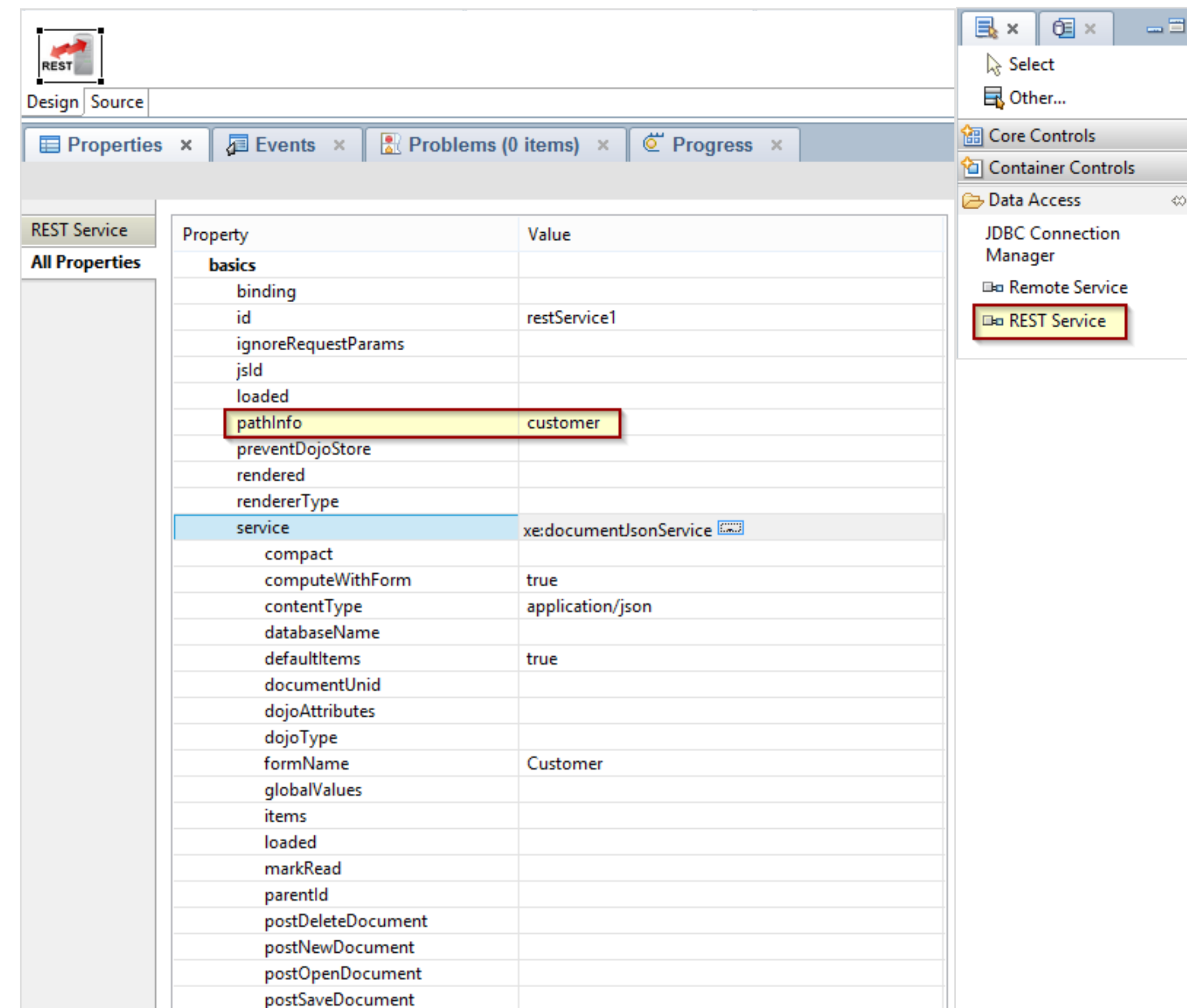


- Direkter Zugriff auf die interne Struktur
- Begrenzt auf Daten in Ansichten oder selektiertem Document
- Eingabevalidierung basiert @Formelsprache
- Eingabevalidierung muss mittels des Query-Strings aktiviert werden



REST-API mittels Extension Library REST-Service

- REST-Service Control ist Teil der Extension Library
- Muss in eine XPage integriert werden
- Zugriff über den Pfad in der Eigenschaft „pathInfo“
 - {database}/{xpage-name}.xsp/{path}



The screenshot displays the SAP NetWeaver IDE interface. The main window shows the 'REST Service' control with its 'All Properties' tab selected. 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'. The right-hand pane shows the 'Data Access' section with 'Remote Service' and 'REST Service' listed. The 'REST Service' is highlighted with a red box.

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



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



Extension Library - xe:customRestService

- Volle Kontrolle über den REST service
- Direkte Implementierung der Methoden
 - doDelete
 - doGet
 - doPost
 - doPut
- Oder „Custom Service Bean“



Extension Library - Custom Service Bean

- Die Klasse muss die Klasse CustomServiceBean erweitern
- Zugriff auf alles was auch eine XPage kann
 - inkl. 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);  
        }  
    }  
}
```



Pro und Kontra



- Unterschiedliche Level der Komplexität
 - Von integrierten Services
 - Bis zur Java implementation
- Mit customRestService volle Kontrolle
- Business-Logik
- Caching



- Gebunden an eine XPage



... und dann wurde es interessant

Sven Hasselbach - XPages: Run your own Servlets

- XPages: Run your own Servlets
 - <http://hasselba.ch/blog/?p=746>
- Verwendung einer ServletFactory von der XPages Runtime um Servlets zu erzeugen
- Basierend auf einem chinesischen IBM Developerworks Blogeintrag
- <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](#)”。

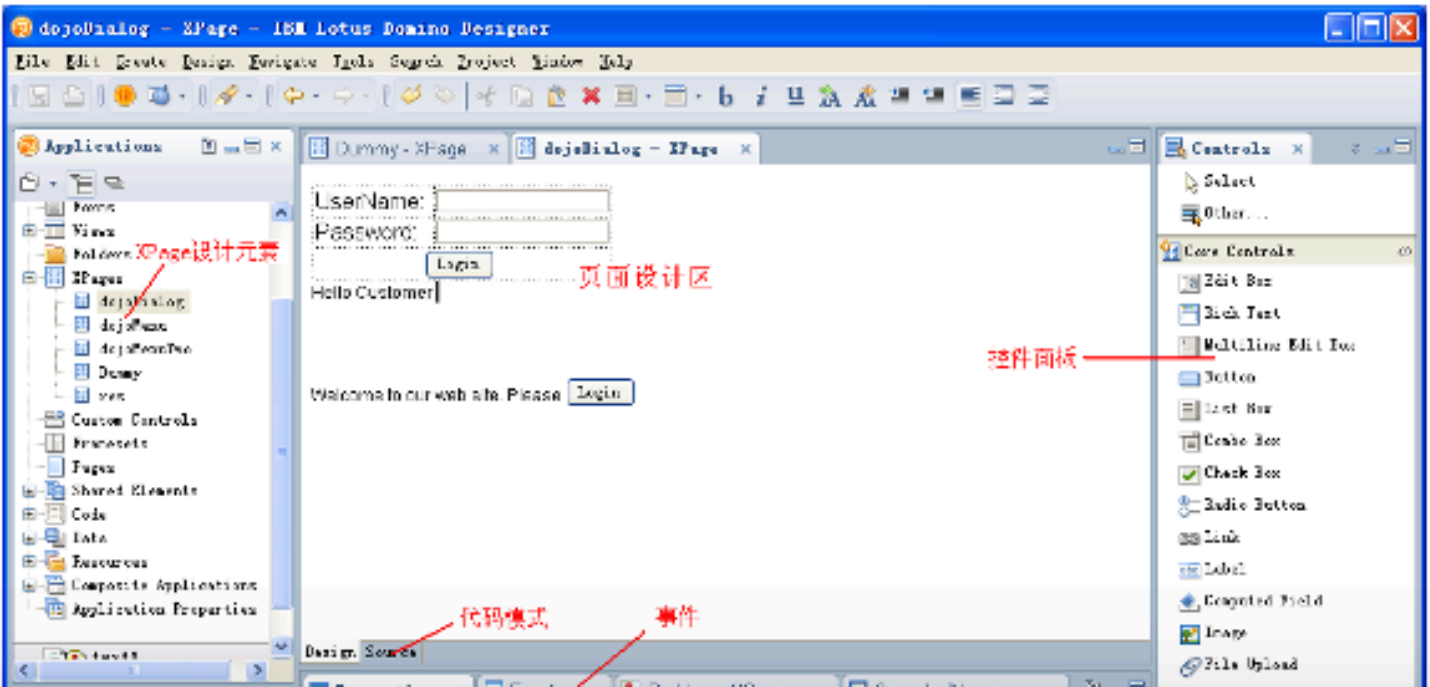
XPage 的前端采用 JSF（Java Server Face）技术实现，并封装了对 Dojo 控件的调用，与传统的 Lotus Domino Web 开发相比较，使用 XPage 开发的 Web 应用界面更加美观。当在 XPage 应用中创建基于 Ajax 的 Web 2.0 应用时，可以采用 XPage 控件及 Dojo 控件构建美观大方的前端界面。但是，还需要一个后端程序来处理业务逻辑，从而实现业务逻辑层及应用显示层的相分离。而 Servlet 在处理业务逻辑上功能强大，是个不错的选择。

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



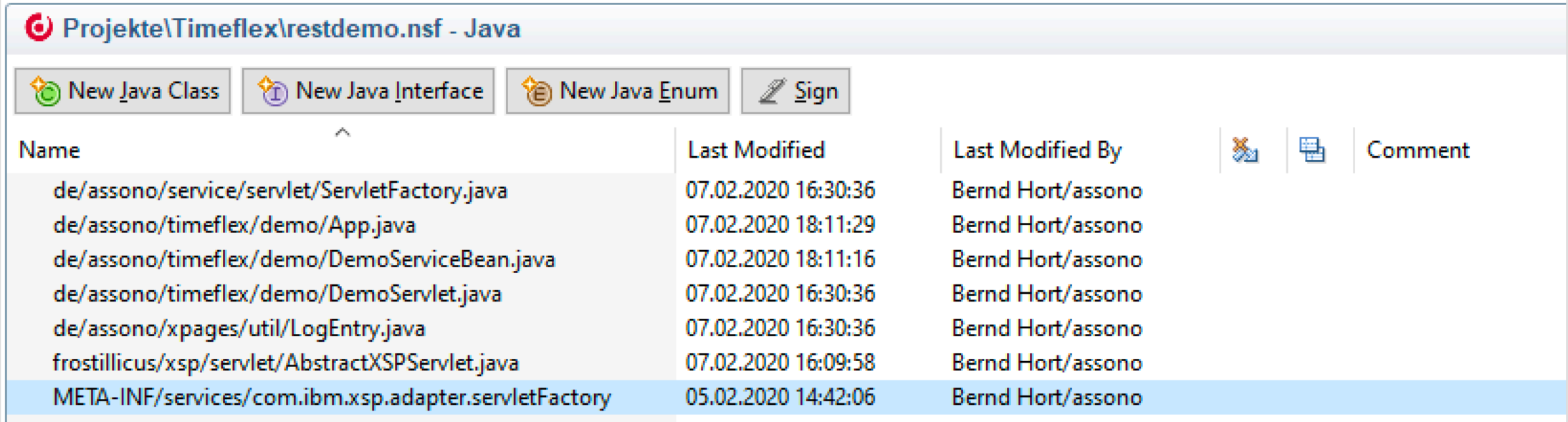
Lotus Domino Designer 8.5.1 免费软件下载

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](#)

Die Servlet Factory

- Die Servlet Factory wird registriert mittels einer Textdatei
 - Der Name muss `com.ibm.xsp.adapter.servletFactory` lauten
- In der Textdatei ist der Java- Klassen-Name der Servlet Factory
- Die Datei muss im Ordner `META-INF/services/` sein
- Der Ordner muss sichtbar für den Java-Build-Path sein
- Ableitung der 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
 - <https://frostillic.us/blog/posts/159496067A27FD3585257A70005E7BC1>
- Basierend auf dem Blogeintrag von Sven Hesselbachs blog und Ergänzungen von Nathan Freeman und Niklas Heidloff beschreibt er wie die Ableitung von DesignerFacesServlet genutzt werden kann, um auf den FacesContext zuzugreifen.
- Mit dem FacesContext kann auf die aktuelle Notes session und die aktuelle Datenbank zugegriffen werden
- Implementierte ein AbstractXSPServlet als Einstiegspunkt für eigene Servlet Implementierungen
- Nicht möglich SessionAsSigner zu benutzen (Class-loader-Probleme)



Pro und Kontra



- Alle Java Features
- Volle Kontrolle
 - Business-Logik
 - Caching
- Deployment mittels Notes-Schablone



- Low-Level Servlet Implementierung
- Kein SessionAsSigner



REST-API mit Java (JAX-RS) mittels Jersey

- Die Java API for RESTful Services (JAX-RS) ist Teil der Java Spezifikation
- Die Sun / Oracles Reference Implementierung Jersey beinhaltet alle notwendigen Klassen
 - Jersey Version 1.71.1 ist die letzte stabile Version, die Java 6 und JAX-RS 1.1 unterstützt
- Erweiterung des ServletContainers und Application von Jersey initialisiert die JAX-RS Application



JAX-RS

- Höheres Abstraktionsebene
- Input Parameter und Rückgabewerte sind Java Objects
- Ein Objekt-Mapper konvertiert Json-Objekte in Java Objekte und umgekehrt
 - Jackson ist eine weit verbreitete Implementierung



Java (JAX-RS) Beispiel



```
@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());
        }
    }
    [...]
}
```



Pro und Kontra



- Alle Java Features
- Volle Kontrolle
 - Business-Logik
 - Caching
- Deployment mittels Notes-Schablone
- Verwendung von JAX-RS



- Kein SessionAsSigner



OSGi Plugins



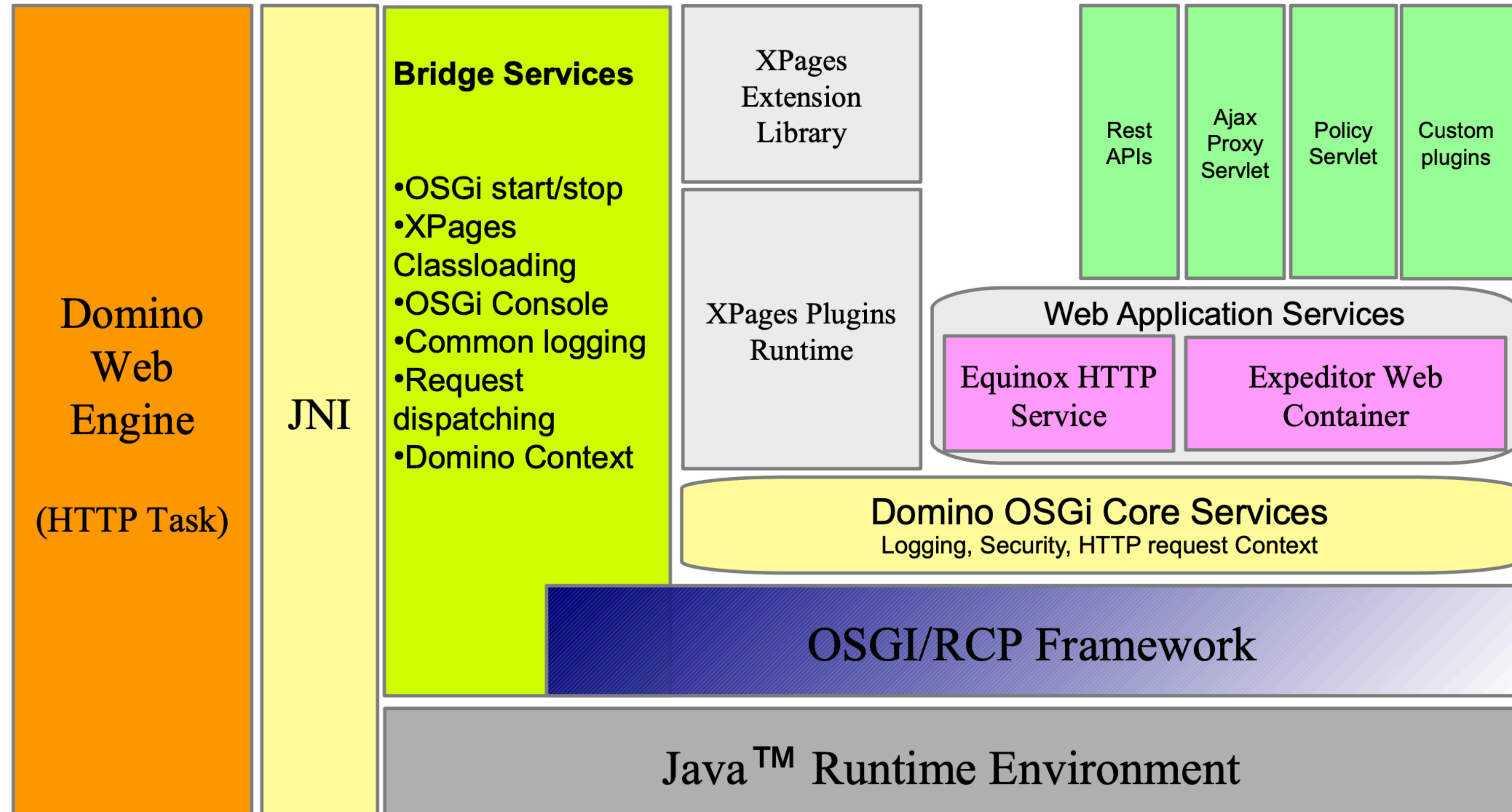
OSGi Plugins

- Seit der Version 8.5.2 gibt es eine OSGi Integration im HTTP task in Domino
- Sie ist die Basis der Extension Library, DOTS und weiteren Erweiterungen
- Paul Fiore (et al) hatte eine Session während der Lotusphere 2012
SHOW112 - Domino OSGi Development
 - <https://www.slideshare.net/fiorep/domino-osgi-development>





IBM Lotus Domino OSGi in HTTP architecture



Web Application Services

- Zwei Komponenten können für eine REST API verwendet werden
- Equinox HTTP Service
 - Für einfachere Servlets
- Expeditor Web Container
 - Basierend auf J2EE



OSGi - Deployments

- Datei basiert
 - Bundles werden als Jars gepackt oder als komplettes Verzeichnis im Domino-Serververzeichnis abgelegt
 - Empfohlener Ort ist {dominodata}/domino/workspace/applications
 - Bundles werden mit maximalen Sicherheitsrechten ausgeführt
- NSF based
 - Bundles werden in eine Notes-Datenbank importiert welche von der UpdateSite.ntf Schablone erbt
 - Server wird via notes.ini konfiguriert die Inhalte der UpdateSite-Datenbank zu laden
 - Die Dokumente mit den Bundles müssen mit einer privilegierten User-Id unterzeichnet sein, um geladen zu werden. Danach werden sie mit maximalen Sicherheitsrechten ausgeführt.



Toby Samples

- Toby Samples antwortete auf Sven Hasselbachs Blog-Eintrag mit =>
- Jahre später schreibt er eine Artikelserie „JAX-RS or THE way to do REST in Domino“
- Darin verwendet er die Apache Wink implementation von der Extension Library um ein JAX-RS-Servlet zu implementieren



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!

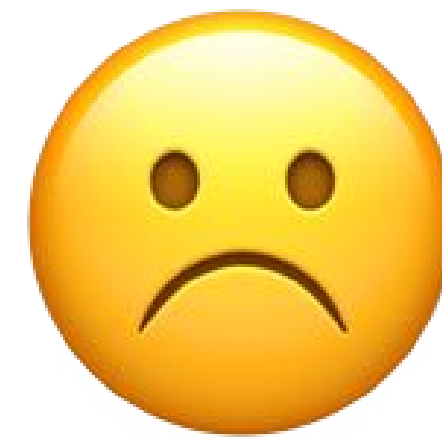
[Antworten](#)



Pro und Kontra



- Alle Java Features
- Volle Kontrolle
 - Business-Logik
 - Caching
- Deployment mittels Notes-Schablone
- Verwendung von JAX-RS
- Maximale Sicherheitsrechte



- Installation des OSGi-Plugin könnte ein Problem sein (Administratoren sind vorsichtige Leute)
- Jedes Update bedeutet ein HTTP-Task-Restart





Photo by [Pedro Sandrini](#) from [Pexels](#)

Ein anderer
Ansatz für Web-
Anwendungen:

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.“
- Die komplette Programmierung basiert in Java - Backend und Frontend



Sven Hasselbach - HowTo: Vaadin on Domino

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



Paul Withers - From XPages to Web App

- Paul Withers veröffentlichte eine exzellente Artikelserie „From XPages to Web App“
 - <https://www.intec.co.uk/from-xpages-to-web-app-introduction/>
- Deckt alles ab von XPages Entwicklung, Verwendung von OSGi bis zur Verwendung von Vaadin um eine Web-Anwendung zu erstellen



Pro und Kontra



- Frontend und Backend werden in Java geschrieben.
- Installation als OSGi-Plugin
=> Maximale Sicherheitsrechte



- Installation des OSGi-Plugin könnte ein Problem sein
(Administratoren sind immer noch vorsichtige Leute)
- Jedes Update bedeutet ein HTTP-Task-Restart
- Notwendigkeit vaadin}> zu lernen



Alternativen zur Domino Java Implementation



OpenNTF Domino API

- Wrapper API für lotus.domino API
- Entwickelt von Jesse Gallagher, Nathan T Freeman, Paul S Withers
- <https://oda.openntf.org/>
- Installiert als ein OSGi Plugin (Domino Server and Domino Designer)
- Enthält Demo Anwendungen und ein Demo-OSGi-Plugins für eine REST API
- Apache License 2.0



Domino JNA

- Cross-platform Zugriff auf IBM/HCL Notes/Domino C API Methoden von Java
- From Karsten Lehmann / Mindoo GmbH
- Github <https://github.com/klehmann/domino-jna>
- Installiert als OSGi Plugin (Domino Server and Domino Designer)
- Apache License 2.0





Business Logik außerhalb von Domino

HCL Domino AppDev Pack

- Das AppDev Pack fügt Node.js Support für den HCL Domino Server.
 - <https://doc.cwpcollaboration.com/appdevpack>
- Drei Komponente
 - Node.js Modul **domino-db** - Zugriff auf Domino Daten von Node.js
 - **Proton** Domino Task - Reagiert auf Anfragen von domino-db
 - Node.js basierter Service **IAM** (Identity and Access Management) für OAuth 2.0 Authentication

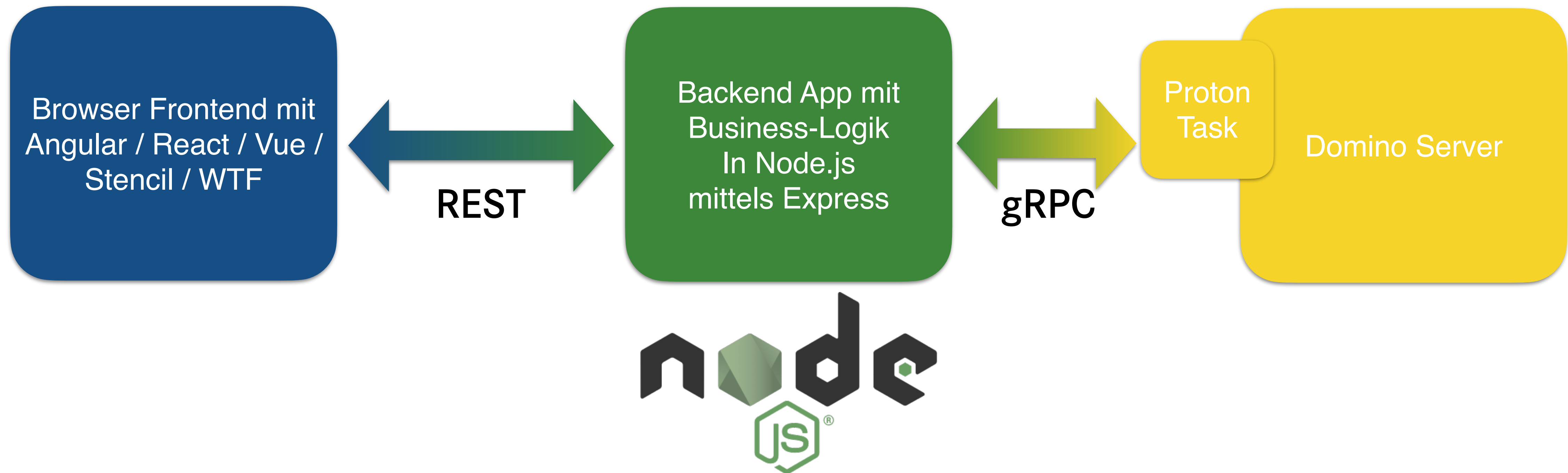


HCL Domino AppDev Pack

- Aus Sicht von Node.js ist Domino nur eine weitere NoSQL-Datenbank
- domino-db ist der Datenbank-Treiber
- HCL löst mit dem Domino AppDev Pack zwei Probleme
 1. Attraktiv für neue Entwickler
 2. Unterstützung eines modernen HTTP-Servers mit HTTP2, WebSockets etc. ohne die Notwendigkeit den HTTP Server in Domino zu aktualisieren



HCL Domino AppDev Pack



Pro und Kontra



- Verwendung von JavaScript für Backend Implementierung
- Express ist ein sehr mächtiger und performanter HTTP Server
- Millionen von Open-Source-Projekten, Tutorials, Tools etc.
- Nutzen der Domino Security und Datenspeicherung



- Domino ist nur Datenspeicher



HCL Domino AppDev Pack - Ressourcen

- Kostenloser Kurs zum Domino AppDev Pack von der C3UG
- <http://www.c3ug.ca/education>
- Google Classroom mit dem Code „feilrb4“



HCL Domino AppDev Pack - Ressourcen (2)

- Projekt von Andi Kress „Open-Workspace“
- AppDev Pack als Backend
- Vue.js als Frontend
- PWA - Progressive Web Application für Offline-Betrieb
- Openntf Webinars - Mai 2021
„May OpenNTF Webinar - Overview and Status of Popular or New Projects“



HCL Domino REST API - Secure access to Domino Design and Data

EARLY ACCESS
BUGS EXPECTED

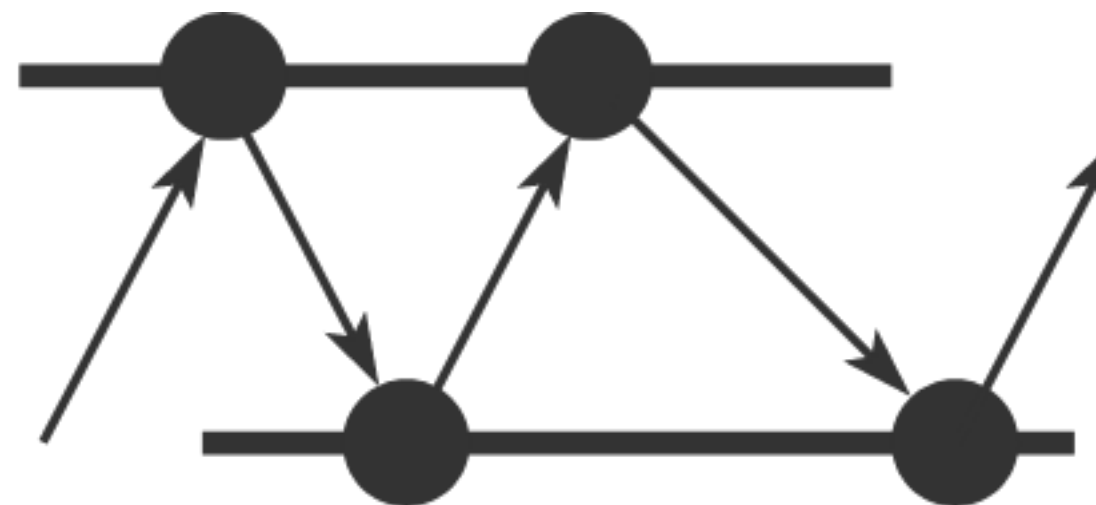
- REST-APIs zum Zugriff auf Domino Design und Daten
 - <https://opensource.hcltechsw.com/domino-keep-docs/>
 - <https://opensource.hcltechsw.com/domino-keep-tutorials/>
- Sehr schnell
- Basiert auf Eclipse Vert.X
 - Integriert seinen eigenen HTTP-Server (incl. HTTP2)
- Verwendet Domino JNX (Java Next Domino)
- Konfiguriert den Zugriff bis hin zur Feldebene
- Datenbank-ACL definiert immer noch die Zugriffsrechte ein Anwender hat
- Unterstützt lokale User



Eclipse Vert.x



- Eclipse Vert.x ist ein Tool-kit für das Erstellen von reaktiven Anwendungen auf der JVM.
- Eclipse Vert.x ist Ereignisgesteuert und Nicht-Blockierend. Das bedeutet eine Anwendung kann eine Vielzahl von Anfragen mittels einer kleiner Anzahl von Kernel-Threads bedienen. Vert.x skaliert sehr gut mit minimaler Hardware.



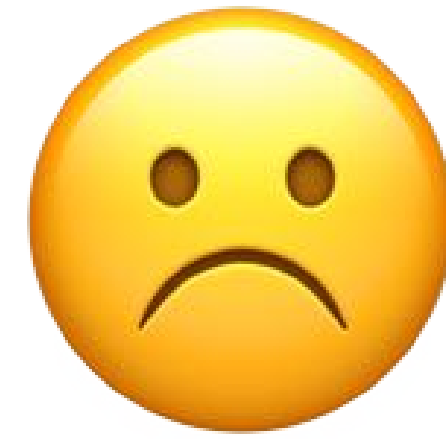
Project Keep



Pro und Kontra



- REST-API für jeden Aspekt einer Domino Datenbank
- Alles konfigurierbar
- Entwickelt im Hinblick auf Sicherheit
- Application-Server kann jeder beliebige Server sein



- Domino ist nur Datenspeicher





Fragen?

Picture Credits

- Unplash

<https://unsplash.com/>

- Pexels

<https://www.pexels.com/>



Stories matter



Di. 29.03.2022 - EntwicklerTrack 1 - 10:15 Uhr