

# GWT pro pokročilé

MoroSystems, s.r.o.

Pavel Klobása

Senior Java Developer

[pavel.klobasa@morosystems.cz](mailto:pavel.klobasa@morosystems.cz)

# Obsah

Co je to GWT?

Návrhové vzory a užitečné knihovny

Komunikace klient - server

ExtGWT

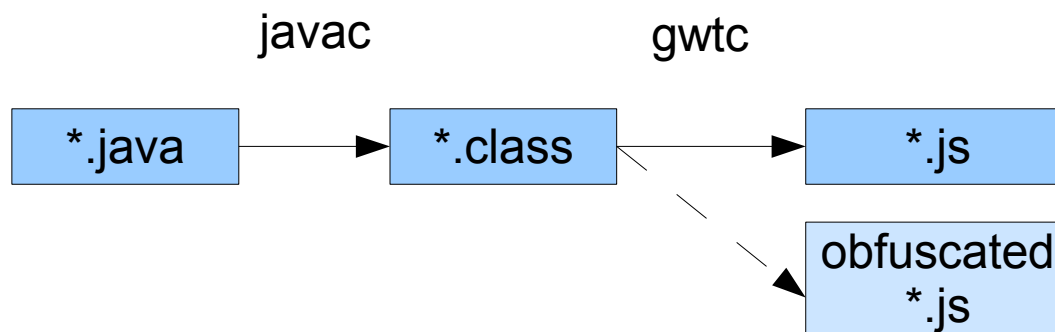
GWT+Spring Roo

# JavaScript je ZLO...

- Slabě typovaný neobjektový jazyk
- Málo datových typů
- Slabá modularizace
- Běhové prostředí ani jazyk není paralelní

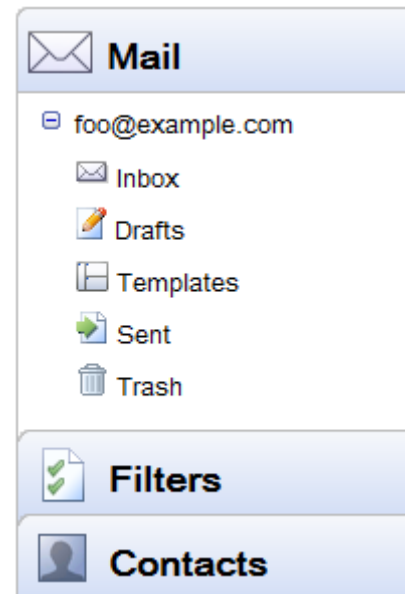
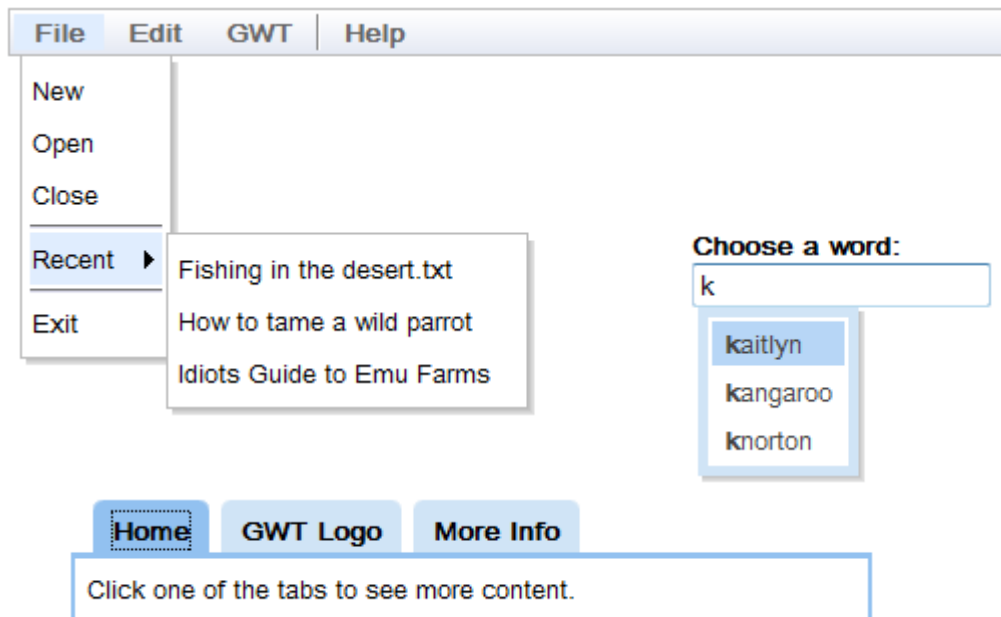
*...JavaScript je v téměř každém prohlížeči...*

# Co je to GWT?



GWT je kompilátor z Javy do JavaScriptu

# Co je to GWT?



GWT je sada komponent

# Co je to také GWT?

- Simulace základních knihoven v JS
- Základní knihovny (DOM, Cookies...)
- RPC - asynch. komunikace se serverem
- i18n
- Skrytí rozdílů mezi prohlížeči
- Události

# Proč GWT? Protože Java...

- Dědičnost, kolekce, interface, generické datové typy...
- Podpora v Eclipse (syntax highlighting, autocomplete, formátování...)
- Maven (repozitory, building)
- Debugging
- FindBugs, CodePro a další...

# Nevýhody GWT

Některé javové knihovny nejsou dostupné:

- Calendar, SimpleDateFormat...
- Reflection, anotace, Class.newInstance()

Zůstává:

- Asynchronní paradigma AJAXu
- Kód neběží paralelně



# ...ale něco tomu chybí..

- Dependency Injection?
- MVC a jiné návrhové vzory?
- Log4J?
- REST?
- Elegantní GUI komponenty?

# Část druhá: Návrhové vzory a knihovny

# Logování pomocí GWT-log

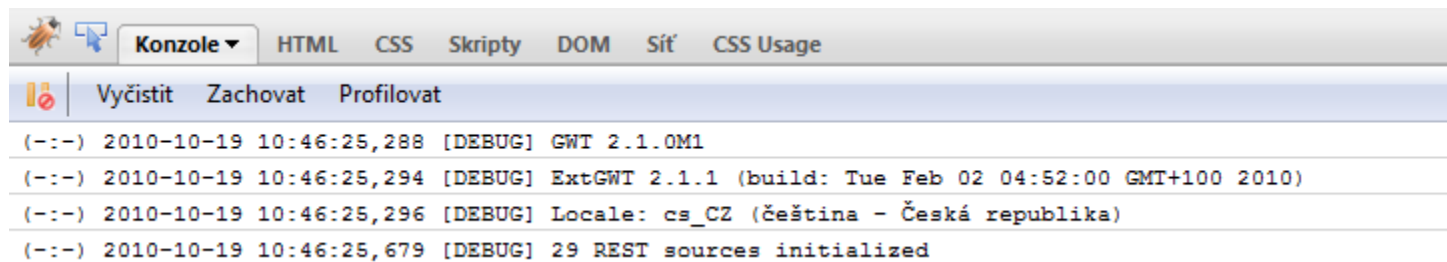
Podobné Log4J:

```
Log.debug("spouštím výpočet...");
```

ConsoleLogger – do konzole prohlížeče

DIVLogger – do HTML stránky

RemoteLogger – na server pomocí GWT RPC



```
Konzole HTML CSS Skripty DOM Síť CSS Usage  
Vyčistit Zachovat Profilovat  
(-->) 2010-10-19 10:46:25,288 [DEBUG] GWT 2.1.0M1  
(-->) 2010-10-19 10:46:25,294 [DEBUG] ExtGWT 2.1.1 (build: Tue Feb 02 04:52:00 GMT+100 2010)  
(-->) 2010-10-19 10:46:25,296 [DEBUG] Locale: cs_CZ (čeština - Česká republika)  
(-->) 2010-10-19 10:46:25,679 [DEBUG] 29 REST sources initialized
```

# Návrhový vzor Event Bus

## Tradiční pojetí

Příjemce se registruje u producenta

## Event Bus

Příjemce se registruje na sběrnici a přijímá všechny události odpovídajícího typu

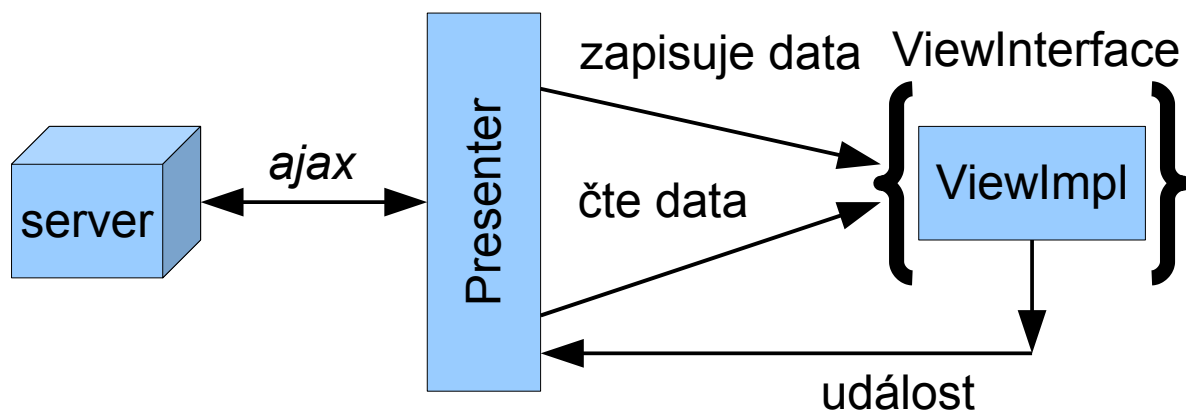
## Výhody EventBus

- Pro registraci nepotřebujeme znát přímý odkaz na instanci producenta
- Lze mít více producentů

# MVP - Model-View-Presenter

- Analogie MVC, ale View je „chytré“ (má stav, interaguje s uživatelem)
- Presenter připraví data a vloží je do View
- View posílá presenteru požadavky uživatele jako události presenteru („ulož data“)

# MVP - Model-View-Presenter



# Dependency Injection - GIN

- Interface Injector – factory pro ostatní případy
- @Inject na konstrukturu – parametry budou injektovány
- Životní cyklus Prototype i Singleton
- Styl interface-implementace

# DI+MVP+EventBus - ukázka

```
public interface TitleEditorView {  
    void setTitle(String title);  
    String getTitle();  
}  
  
public class SaveEvent {  
    .....  
}  
  
public interface SaveEventHandler {  
    public void onSaveTitleRequest(.... event);  
}
```



# DI+MVP+EventBus

```
public class TitleEditorPresenter {
    @Inject
    public TitleEditorPresenter(final DataProvider provider,
                               final TitleEditorView view, EventBus bus){
        provider.load(new DataLoadHandler(){
            public void onLoad() {
                view.setData(provider.getTitle());
            }
        });
        bus.addHandler(SaveEvent.TYPE, new SaveEventHandler() {
            public void onSaveData(SaveEvent ev){
                provider.saveTitle(view.getTitle());
            }
        });
    }
}
```

# DI+MVP+EventBus

```
public class TitleEditor extends VerticalPanel implements TitleEditorView{
    TextBox field;
    @Inject
    public TitleEditor(final EventBus bus){
        field=new TextBox();
        add(field);
        Button saveBtn = new Button("Ulož");
        saveBtn.addClickHandler(new ClickHandler() {
            public void onClick(final ClickEvent event) {
                bus.fireEvent(new SaveTitleRequestEvent());
            }
        });
        add(saveBtn);
    }
    public void setTitle(String title){
        field.setText(title);
    }
    public String getTitle(){
        field.getText();
    }
}
```

# Část třetí: Data

# Komunikace server - prohlížeč

GWT RPC jako zdroj dat z databáze?

- Stáhne na klienta všechny atributy entity včetně navázaných entit
- Problém s Hibernate Lazy asociacemi
- Na serveru musí být JVM+naše aplikace
- Entity musí být `Serializable/Serializable`
- Na serveru máme jiné požadavky na data než na klientu (DTO)

# JSON

- Fragment JS kódu
- Faktický i formální standard
- String, Number, Array, Object/Map, NULL

# REST

- Bezstavová komunikace
- Každá služba (a každý objekt) vlastní URL

*REST API definuje rozhraní server-klient*

# JSON na serveru - org.json

```
java.io.Writer wr=...
JSONWriter writer=new JSONWriter(wr);
writer.array();
for(Product prod:products){
    writer.object();
    writer.key("id").value(prod.getId());
    writer.key("name").value(prod.getName());
    writer.endObject();
}
writer.endArray();
```

# JSON na serveru - GSON

```
Gson gson=new Gson();  
String outputJson=gson.toJson(myObj);
```

@Expose, ExclusionStrategy – potlačení atributu  
JSONSerializer<T> – vlastní konverzní předpis

# JSON na klientu

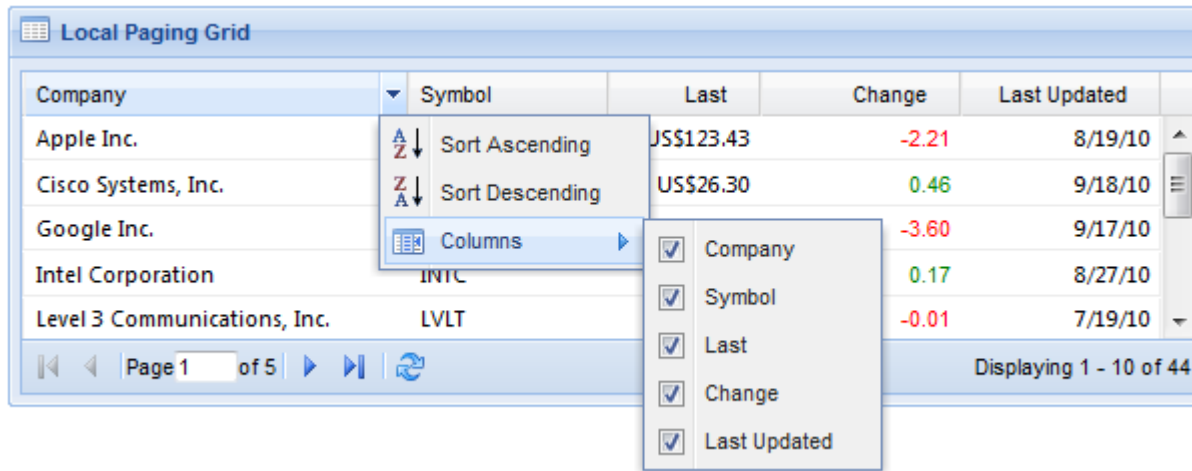
*Jak naplnit bean daty bez reflexe:*

```
public class MyEntity{
    protected Map<String,Object> data=
        new HashMap<String,Object>();
    public void setId(Integer id){
        data.put("id", id);
    }
    public Integer getId(){
        return (Integer)data.get("id");
    }
}
```



# Část čtvrtá: ExtGWT

# Datový grid v RIA...



The screenshot shows a web application window titled "Local Paging Grid" containing a table of stock data. The table has columns for Company, Symbol, Last, Change, and Last Updated. A context menu is open over the "Symbol" column, offering "Sort Ascending" and "Sort Descending" options. Another menu is open over the "Columns" header, showing a list of columns with checkboxes: Company, Symbol, Last, Change, and Last Updated. The table data includes Apple Inc., Cisco Systems, Inc., Google Inc., Intel Corporation, and Level 3 Communications, Inc. The footer of the grid shows "Page 1 of 5" and "Displaying 1 - 10 of 44".

Company	Symbol	Last	Change	Last Updated
Apple Inc.		US\$123.43	-2.21	8/19/10
Cisco Systems, Inc.		US\$26.30	0.46	9/18/10
Google Inc.			-3.60	9/17/10
Intel Corporation	INTC		0.17	8/27/10
Level 3 Communications, Inc.	LVLT		-0.01	7/19/10

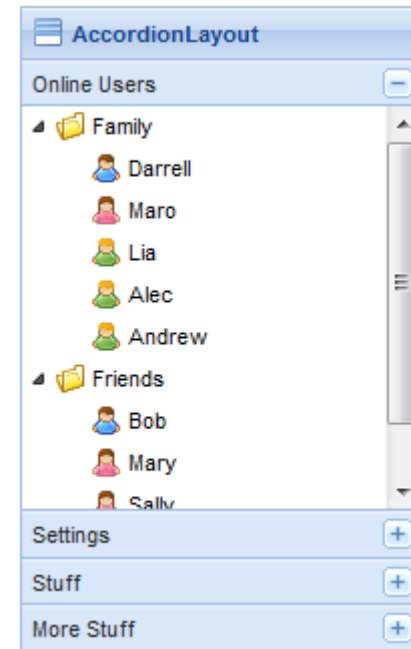
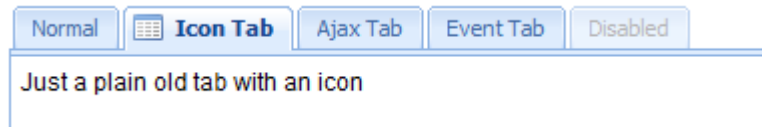
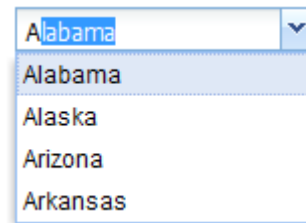
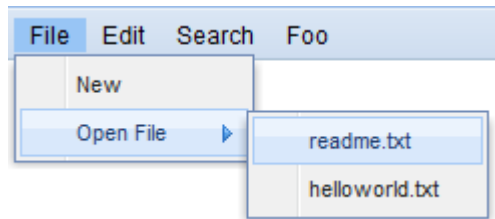
## Požadavky:

Stránkování, dotahování dat ze serveru, změna řazení, změna pořadí sloupců, skrývání sloupců, formátování sloupců a políček, inline editace políčka...

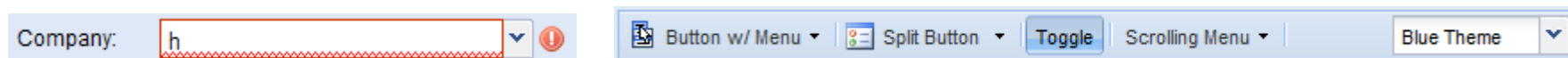
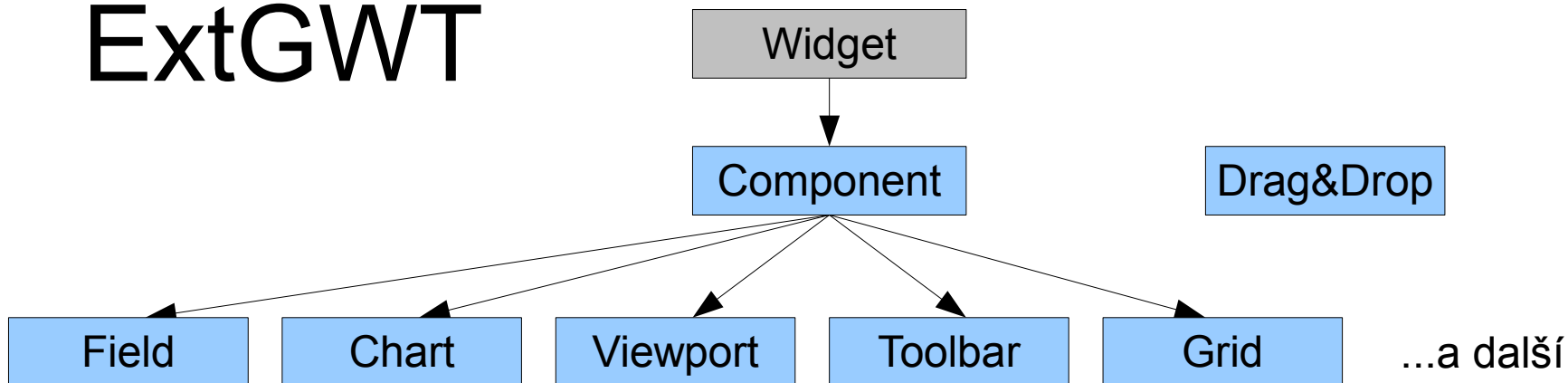
# ExtGWT vs. SmartGWT

- GPL/commercial
- 99% kódu v Javě
- Přepsané ExtJS
- LGPL
- Javové API nad JS
- SmartClient obalený GWT
- Komponenty lze mixovat s GWT komp.
- Vlastní implementace datových zdrojů
- Placená podpora

# ExtGWT – ukázky komponent



# ExtGWT



Layout

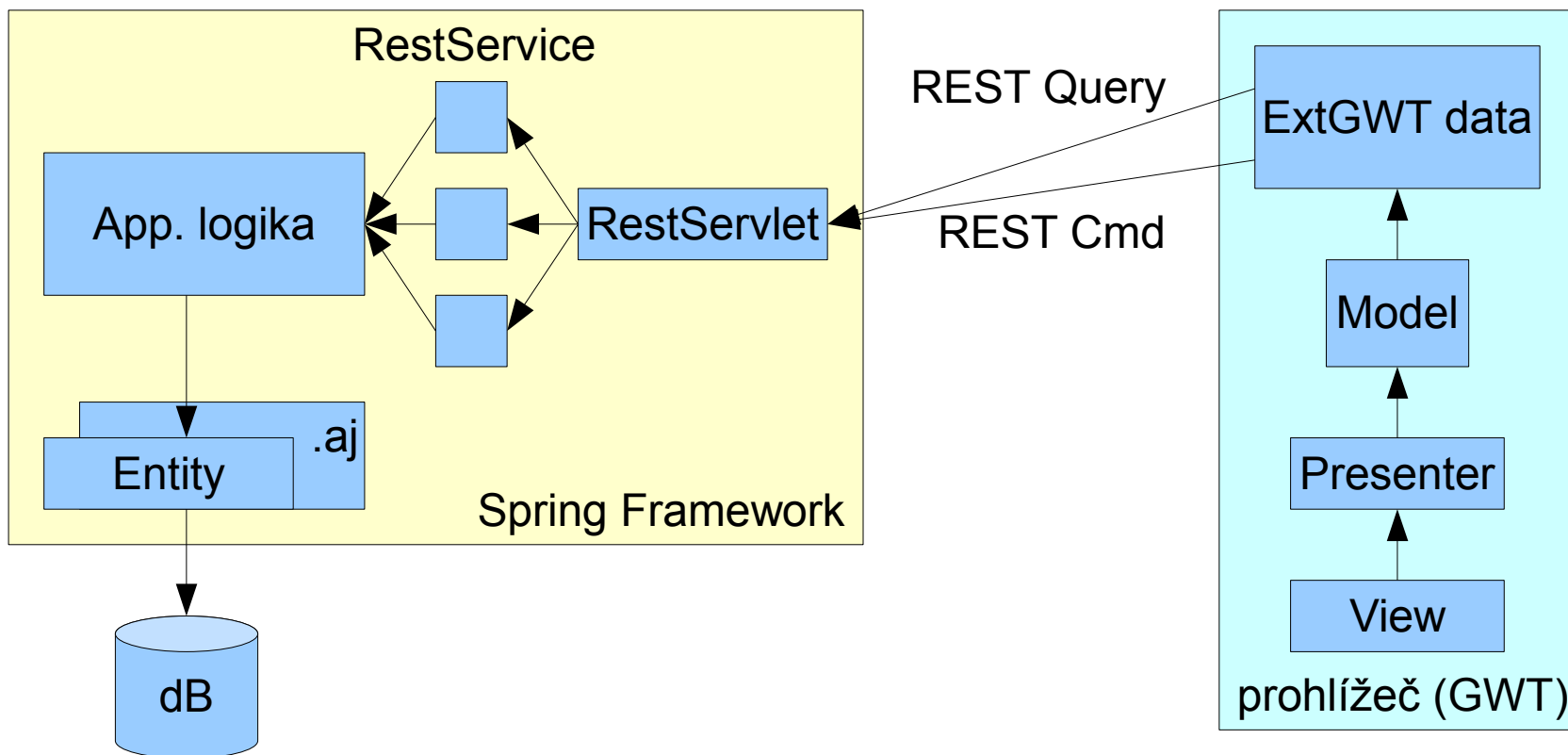
- BorderLayout
- FitLayout
- TableLayout
- ...a další

Data

*Návrh GUI: GWT Designer nebo Ext Designer.*

# Část pátá: Spring Roo

# GWT+Spring Roo dohromady



# Další informace:

<http://code.google.com/intl/cs/webtoolkit/>

<http://code.google.com/p/gwt-log/>

<http://www.sencha.com/products/gwt/>

<http://code.google.com/p/google-gin/>

<http://www.json.org>

<http://code.google.com/p/google-gson>

<http://www.youtube.com/watch?v=PDuhR18-EdM>