Overview

- Java developers have had issues with how to effectively integrate GIS maps and data into their web applications for many years.
- In addition, most Java developers are not as attuned to graphic and user interface (UI) design as graphic artists and UI specialists.
- To solve this problem on the user presentation side, Java Server Faces (JSF) was developed and several third-party component libraries, such as PrimeFaces, OpenFaces, ICEFaces, were developed to help solve this problem.
- The same is true for map integration into Java web applications and that is why GISFaces was created.
- This discussion explains how these tools are combined to provide incredible enterprise applications for clients.
Topics for Discussion

• What is GISFaces?
• Simple Example
• Why the need?
• What are the features?
• Why use the ESRI ArcGIS API for JavaScript?
• Why use JavaServer Faces?
• Advanced Example
• What are the requirements?
• What is next?
• How to get more information about GISFaces?
• Questions
What is GISFaces?

- Java Enterprise Edition (EE) component library for GIS mapping.
- Core technologies used are the ESRI ArcGIS API for JavaScript and JavaServer Faces.
- Library includes reusable components which hide low level details of web mapping.
- Provides tight coupling between Java and GIS in a web environment.
- JSF web pages are created using a declarative xhtml markup.
Simple Example - Map
Simple Example - Code

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
     xmlns:f="http://java.sun.com/jsf/core"
     xmlns:h="http://java.sun.com/jsf/html"
     xmlns:gis="http://gisfaces.com">
  <h:head>
    <title>GISFaces Example Map</title>
  </h:head>
  <h:body>
    <h:form>
      <h:panelGroup id="panel1" layout="block" styleClass="gisfacesMapPanel"></h:panelGroup>
      <gis:map mapPanel="panel1" background="streets" latitude="39.826175" longitude="-98.5795" zoom="4"></gis:map>
    </h:form>
  </h:body>
</html>
```
Why the need for GISFaces?

- No product currently exists that leverages these technologies.
- The ESRI Java Web ADF is deprecated.
- Reusable library that hides the complexity of the interaction between JSF and the ESRI ArcGIS API for JavaScript.
- Allows any page author or developer to easily create a GIS map in a web page.
- No need to be a Java, JavaScript, ESRI JSAPI, or GIS expert.
What are the features of GISFaces?

- Uses the ESRI ArcGIS API For JavaScript mapping engine.
- Supports ESRI tiled, image, and dynamic map services.
- Supports KML, feature, and graphics layers.
- Supports the 10 standard ESRI tiled backgrounds.
- Graphics layers support marker, polyline, polygon, circle, and text graphics.
- Graphics layer markers support drag and drop functionality.
- Ability to add multiple services and set opacity, refresh interval, and min/max scales per service.
- Dynamic map services support layer visibility and definition expressions for filtering.
What are the features of GISFaces? (Continued)

• Support for legend, overview, navigation, and geocoder widgets in separate panels.
• Options to show a scalebar, logo, attribution, latitude/longitude coordinates, and a progress bar in separate panels.
• Supports <f:ajax> events “click”, “extent”, “view”, “select”, “drag”, and “geolocation”.
• Supports feature identification, highlight, and attribute table via <f:ajax> listener.
• Supports custom map LOD (level of detail) levels and layer min/max scale ranges.
• Supports automatic map layer refreshes at specified intervals.
• Ability to use a locally hosted ESRI JSAPI for environments behind a firewall or without Internet access.
Why use the ESRI ArcGIS API for JavaScript?

- The ESRI ArcGIS API for JavaScript is a feature-rich, browser-based mapping engine.
- Based on JavaScript, not third-party plugins, which are issues within some organizations and governments.
- It is a fast, lightweight, and simple library.
- Uses Open Standards such as JavaScript, REST, and JSON.
- Has a broad target audience in terms of device and browser support.
- The website is [https://developers.arcgis.com/javascript/](https://developers.arcgis.com/javascript/).
Why use JavaServer Faces?

- JSF technology greatly simplifies building web pages.
- Is the de facto standard in Java web development.
- Component-based framework.
- Event driven programming model with AJAX built in.
- Contains a feature rich, reusable set of high and low level components.
- Component model which allows third party development of custom components, like GISFaces.
- Many third party libraries are freely available and include PrimeFaces, RichFaces, IceFaces, OpenFaces, MyFaces.
- The website is https://javaserverfaces.java.net/.
Advanced Example - Map
Advanced Example - Code

```html
<!-- Code snippet -->
```

This is an example code snippet.
What are the requirements to use GISFaces?

- Java 8 or greater.
- JavaServer Faces 2 or greater.
- Java web container such as Glassfish, WildFly, Tomcat, or WebSphere.
- Access to the ESRI ArcGIS API For JavaScript API via Internet or local hosting.
- Some options may require map services hosted on ESRI ArcGIS Server 10 or greater.
What is next for GISFaces?

- Support graphics layer geometry edits including drag, scale, rotate, and vertices editing for polygons.
- Convert to a dynamic map model defining services and layers instead of declarative XHTML markup.
- Allow custom popup content using a template or JavaScript callback function.
- Considering support for the ESRI/Leaflet mapping engine.
Example Applications
Example Applications
Example Applications
Example Applications
How to get more information about GISFaces?


• Download the .zip file which contains the binary .jar, javadoc .jar, and example .war files.

• Run live examples.

• Please give us your feedback!
Questions and
Thanks for your Attention!
“Bringing Enterprise GIS to Java Server Faces”

William “Pat” Geer
pat@geerservices.com

Chris Duncan
chris@geerservices.com