# Embedding search engine

XFlyGate class have to be used for embedding our flights search engine on Your website. First, XFlyGate.dll reference have to be added to Your project.

## 1. Embedding 1st and 2nd step

a) Agent number, affiliate , GET, POST have to be provided for the XFlyGate class contructor. Last constructor parameter is optional, it indicates that http or https must be used during request to the search engine. This value is set by default to false.

```
Dictionary<string, string> GET = new Dictionary<string, string>();
Dictionary<string, string> POST = new Dictionary<string, string>();
bool useHttp = false;
string agentNumber = YOUR_AGENCY_NUMBER;
string affiliate = YOUR_AFFILIATE;
XFlyGate.XFlyGate xflyGate =
       new XFlyGate.XFlyGate(agentNumber, affiliate, POST, GET, useHttp);
```

b) URL to the search engine have to be set after XFlyGate construction.

```
xflyGate.SetUrl("http://xfly.merlinx.pl/EU/");
```

c) Now, `FetchParts` method can be used for downloading content of search engine to your website.

```
ContentParts contentParts = xflyGate.FetchParts("headCss", "headJs",
       "mxMainForm", "mxResultListLoaderGlobal", "mxLandingPage");
```

d) Css and javascript parts have to be attached in the HEAD section of the website. These parts can be obtained in the following way:

```
string cssContent = contentParts.FetchPart("headCss", false);

string jsContent = contentParts.FetchPart("headJs", false);
```

e) The rest of the search engines components can be placed in selected areas of the website. All elements are optional.

Main parts of the search engine:

//main search engine form(step 1):

```
string sideFormContent = contentParts.FetchPart("mxSideForm");
```

//progress bar visible during search:

```
string loaderContent = contentParts.FetchPart("mxResultListLoaderGlobal");

string landingContent = contentParts.FetchPart("mxLandingPage");
```

//results of search(step 2):

```
string resultContent = contentParts.FetchPart("mxResult")
```

Helper search engine(vertical):

```
string sideFormContent = contentParts.FetchPart("mxSideForm");
```

Filters(vertical):

```
string filtersContent = contentParts.FetchPart("mxFilters");
```

f) It's up to You and framework used in Your project how fetched search engine content will be embedded on your website.

g) To enable steps "back" functionality, set URL of the website with attached 1st and 2nd steps in configuration panel (Agency profile > IBE profile > Subpage with the search engine (GATE version)).

## 2. Embedding 3rd and 4th step

**NOTE:** HTTPS (HTTP protocol with encryption) server have to be used for step 3 (booking form) . This protocol is a must for security reason (Privacy policy e. g. credit card data). If you do not have HTTPS server, request to the booking form will be redirected to MerlinX server by default. If you own HTTPS server 3rd and 4th step can be embedded in your website. For system to work properly URL of the webstie with these steps must be set in IBE configuration panel (Agency profile > IBE profile > Subpage with the booking (GATE version)).

a) Agent number, affiliate , GET, POST have to be provided for the XFlyGate class contructor. HTTPS parameter must be set to true. This value is set by default to false. **Remember to read GET parameters**. `ReadGetParameters method isn't included in library, it's only here as an example.`

```
Dictionary<string, string> GET = ReadGetParameters();
Dictionary<string, string> POST = new Dictionary<string, string>();
bool useHttp = true;
string agentNumber = YOUR_AGENCY_NUMBER;
string affiliate = YOUR_AFFILIATE;
XFlyGate.XFlyGate xflyGate =
       new XFlyGate.XFlyGate(agentNumber, affiliate, POST, GET, useHttp);
```

b) URL to the search engine have to be set after XFlyGate construction.

```
xflyGate.SetUrl("http://xfly.merlinx.pl/EU/");
```

c) Now, `FetchParts` method can be used for downloading content of search engine to your web site.

```
ContentParts contentParts = xflyGate.FetchParts "headCss", "headJs",
       "mxBookingForm", "mxResultListLoaderGlobal");
```

h) `Again,` css and javascript parts have to be attached in the HEAD section of the web site.

```
string cssContent = contentParts.FetchPart("headCss", false);
string jsContent = contentParts.FetchPart("headJs", false);
```

d) The rest of the search engines components can be placed in selected areas of the website. All elements are optional.

```
Main part of search engine:
// booking form (step 3):
```

```
string bookingFormContent = contentParts.FetchPart("mxBookingForm");
// progress bar visible during offer availability checking:
string loaderContent = contentParts.FetchPart("mxResultListLoaderGlobal");
string gLoaderContent = contentParts.FetchPart("mxResultListLoaderGlobal");
```

## 3. 1st and 2nd steps javascript events

You can attach 1st and 2nd steps javascript events to recognize which one of them is currently visibible.

a) 1st step with search form:

```
mxResultHandlers.eventShowMainForm = function () {
    console.log('eventShowMainForm');
};
```

b) Screen “Searching. Please Wait”

```
mxResultHandlers.eventStartSearch = function () {
    console.log('eventStartSearch');
};
```

c) 2nd step with results:

```
mxResultHandlers.eventShowResult = function () {
    console.log('eventShowResult');
};
```

## 4. Example

SampleXFlyGateMvcIntegration is Visual Studio 2013 MVC 4 project. It contains 1st, 2nd, 3rd, 4th embedding and javascript events. To make example work do the following:

a) you have to setup AGENT_NUMBER and AFFILIATE const in SampleXFlyGateMvcIntegration.Models.Agency class.

b) set website in IIS.

c) set Subpage with the search engine (GATE version) and Subpage with the booking (GATE version) as it was described in points 1.g and 2.

Project description:

APP_Start:

BundleConfig – contains sample-script bundle.

RouteConfig – defines routs to search and booking controllers.

Scripts:

sample-script.js – contains attached events of 1st and 2nd steps.

Controllers:

BaseController – defines methods for reading GET and POST parameters.

HomeController – fetches  1st and 2nd steps.

BookingController – fetches 3rd and 4th steps.

Models:

Agency – defines AGENT_NUMBER and AFFILIATE.

ViewModels:

BaseViewModel  - contains Title and ContentParts properties.

SearchViewModel  - it’s derived from BaseViewModel . Model for Home view.

BookingViewModel  - the same as SearchViewModel  but used in Booking view.

Views:

Home > Index.cshtml  - has embedded parts fetched from  1st and 2nd steps.

Booking > Index.cshtml  - contains fetched parts for 3rd and 4th steps.

Shared > _Layout.cshtml  - main layout with head and scripts section.