XContest API implementation into 3rd party websites

Basics

Basic use of XContest API is very simple. You must:

1. **include API javascript library** into your HTML code
2. **create HTML container** (div) for displaying API service
3. **call API service** you want directly in javascript code after API library inclusion

API javascript library inclusion

Put this code snippet inside HTML code of your page:

```
<script type="text/javascript"
src="http://www.xcontest.org/api/js/?key=YOUR-XCONTEST-API-
KEY"></script>
```

But attention please: YOUR-XCONTEST-API-KEY must be replaced by your own **XContest API key**. If you don't have any, ask us.

This code snippet should be in HTML code only **once**. It must be placed **before** any javascript with API service call (see third step) in source code flow.

Creation of HTML container

Just place **div** element somewhere into your HTML code, where you like to display API service.

```
<div id=”flights”></div>
```

Div must have **id attribute** according to the API service you like to show. In this example you like to show **flights** list.
This code can be placed anywhere in your HTML code, but it must be placed **before** javascript with API service call (see next step) in source code flow.

Calling API service

In our example we like to show flights. The minimalistic code can be following:

```
<script type="text/javascript">

XContest.run("flights", {
      source : {
           league : "cpp",
           volume : "2010",
           subcontest : "krupka"
      }
});
</script>
```

As you can see, first argument for XContest.run() method is an **indentifier of API service** and it corresponds to the value of **div's id attribute** in second step. Second argument is well-known javascript object structure with various **settings**. The only mandatory part of the settings object is **source** with the definion of contest you need to display.

This code displays table with last 50 flights in **Krupka** subcontest of **CPP** (Czech Paragliding Cup) for season **2010**.

Available services

Except **flights** service, it is possible to use many other services. Here is a list of them.

name	**Identifier** **(for Xcontest.run() in Javacript)**	**Default id** **(attribute value for HTML container)**
Flight list	flights	flights
Flight detail	flight	flight
Pilot list	pilots	pilots
Pilot detail	pilot	pilot
Ranking in category	ranking	ranking

Changing default container id

It is possible to use different id of div used as API service container. If you do so, you must specify container for current API service as a **third argument** of run() method:

```
<div id="flights-of-my-club"></div>


<script type="text/javascript">
XContest.run("flights", {
      source : {
           league : "cpp",
           volume : "2010",
           subcontest : "krupka"
      },
      "flights-of-my-club"
});
</script>
```

Alternatively, you can pass also DOM node instead of id value as a third argument:

```
<div id="flights-of-my-club"></div>


<script type="text/javascript">
XContest.run("flights", {
      source : {
           league : "cpp",
           volume : "2010",
           subcontest : "krupka"
      },
      document.getElementById("flights-of-my-club")
});
</script>
```

Common settings

All of the widgets supports following settings in settings object.

Language selection

By default, XContest API try to find language used on current page – simply reads value of **lang** attribute of root **html** tag – if the attribute is found and the language is avaible on XContest, API uses such language for all language-dependent texts. In case this process fails, API uses **english (en)** as default.

You can also enforce using of defined language in javascript code as a part of settings **global.lng**. See example:

```
<script type="text/javascript">

XContest.run("flights", {
     global : {
          lng : "de"
     },
      source : {
          league : "cpp",
          volume : "2010",
          subcontest : "krupka"
     }
});
</script>
```

API uses German (de) language now.

You can use these languages:
ca català, **cs** čeština, **de** deutsch, **en** english, **es** espagnol, **fi** suomi, **fr** francais, **hu** hungarian, **it** italiano, **lt** lithuanian, **lv** latvian, **nl** nederlands, **pl** polski, **ro** română, **sk** slovenčina

Providing Google Maps API key

To make Flight list, Flight detail and Pilot detail fully functional, you must provide **Google Maps API key** for your website, because XContest API uses Google Maps API for map mashup and also for quick map in flight list. If you don't have any key, you should create it here http://code.google.com/intl/en/apis/maps/signup.html

Then you must provide it through **global.googleMapsApiKey** in settings:

```
<script type="text/javascript">

XContest.run("flights", {
     global : {
          googleMapsApiKey : "YOUR-GOOGLE-MAPS-API-KEY",
          lng : "de"
     },
      source : {
          league : "cpp",
          volume : "2010",
          subcontest : "krupka"
     }
});
</script>
```

Of course, string YOUR-GOOGLE-MAPS-API-KEY must be replaced by your own one (looks for example like this:

```
ABQIAAAAJRTrjKixiUbwbvD20Z9hvhTgUWx3jL4OAr2-xwZeXYN40aBW9hSXJ_ot3xvCKKJtYKKmoWAwySVHMw
```

Binding with other pages

By default, any link from XContest API service goes to XContest server. For example in flight list, each row contains link to flight detail and pilot detail – both goes to concrete page on XContest server (so away from your website) by default.

If you like to change this behavior, you must fullfil two conditions:

1. create another page which shows a target of the link (for example, general page for displaying flight detail)
2. add rules for binding into settings section

Important: As you can see, usually you are not preparing page for an explicit item (for example one concrete flight or one concrete pilot). Instead, you are preparing page which is able to display any flight from your contest, based on hash part of page URL (hash is a part of URL after # sign).

So let's show it on example. First we must initialize another page with **flight detail** API service. URL of such page will be for example **www.my-pg-page.org/contest/flight** and initialization inside HTML code should look like this:

```
<div id="flight"></div>


<script type="text/javascript">
XContest.run("flight", {
      source : {
           league : "cpp",
           volume : "2010",
           subcontest : "krupka"
      }
});
</script>
```

Now, we must return back to the page with **flight list**. There are new properties in the settings:

```
<script type="text/javascript">
XContest.run("flights", {
      source : {
           league : "cpp",
           volume : "2010",
           subcontest : "krupka"
      },
      sitemap : {
           flight: "/contest/flight"
      }
});
</script>
```

Group **sitemaps** contains links to other types of pages of your contest – just now **flight** and **pilot** (other page types of pages will be added in the future).

Other settings - principle

Each API service has specific range of settings, but there are some common groups of rules.

Let's look at complete default settings of **flight** API service and focus on new yellow backgrounded code:

```
<script type="text/javascript">
XContest.run("flights", {
      global : {
            googleMapsApiKey : "YOUR-GOOGLE-MAPS-API-KEY",
            lng : "de"
      },
      source : {
            league : "cpp",
            volume : "2010",
            subcontest : "krupka"
      },
      has : {
            cols : ["number", "time_start", "pilot", "launch", "route",
"distance", "points", "duration", "speed", "glider", "info",
"show"],//list/array of displayed columns
            filter : true, //show filter?
            pagingTop : true, //show paginator over the list?
            pagingBottom : true, //show paginator under the list?
            header : true, //show header of the table?
            timeStartDate : true, //show time (hh:mm) in start time?
            timeStartTime : true, //show date (dd.mm.yy) in start time?
            pilotCountry : false, //show flag of pilot's country?
            pagingKeypress : true, //move between pages of list with keyboard
arrows?
            roundPoints : false //round points value (no decimal places)?
      },
      list : {
            maxPages : 10, //maximum number of pages shown in paginator
            defaults : {
                  start : 0, //start list from what item? (offset)
                  num : 50, //number of items on 1 page of the list
                  sort : "time_claim", //sort by what column?
                  dir : "down" //sort direction - "down"/ "up"
            },
            sortBy : ["time_start", "pilot", "route", "distance",
"points", "duration", "speed", "glider", "time_claim"] //according to what
columns it is possible to re-order?
      },
      filter : {
            filterVars : ["date", "catg", "country"], //what fields in filter
section?
            joinList : {sort : "reg"} //default sorting after filtering
      }
});
</script>
```

First group called **has** contains everything what can be “switched on/of” - in other words, displayed or not displayed. Typical value for the property in this groups is **true** (displayed) or **false** (not displayed), with only one exception **cols** (array of names of displayed columns of the table).
Second group called **list** contains all settings for paging and sorting of the list. This group is needed only for “list” API services (like Flight list or Pilot list).
Third group called **filter** contains settings for list filter (if present – see **has.filter**).
Meaning of each settings item is explained in the comment (small green text).
Important: You don't need to use (write) these settings explicitly, until you need to set something differently compared to default setting – thus you should insert only properties you like to change.

Customizing default look - CSS

By default, each service is displayed in XContest default style. If you like to change it, or you like to repair conflicts of XContest CSS styles with CSS stylesheet for your page, you can simply add custom stylesheet or custom rules into your own stylesheet.

Complete stylesheet which is used by XContest API is here
http://www.xcontest.org/api/_custom/css/xcontest.css

There are some basic principles which you should know before you start to tweak CSS:

Each root HTML container (div) for API service gets classname **XContest**, and each rule in XContest stylesheet starts with this default classname. Thus CSS rules in default XContest stylesheet may not influence other HTML code outside API service container, until you are not using classname XContest anywhere else in your HTML code.

On the other side, CSS rules in your website's stylesheet may influence (in some cases break) styles for XContest API service. Many general, not specific CSS rules may change usual look of widgets based on XContest API. To solve this issue, you must identify, what rules are in conflict with specific XContest rule and then create more specific rule which begins with .XContest …

To find out what exactly causes conflics in CSS it is very usefull to work with webdesigner extension like FireBug (for Firefox and Chrome browsers) which contains tools for identifiing all CSS rules related to each html element.