

## Project 3: Consume a web service

---

Using the same RESTful service consumed in Project 2 create a C# front-end for this information. Please be careful to include all the 'user experience' tools you have learned to give feedback on what the application is capable of and what a user should do.

### Requirements

---

- Write search functionality to obtain a specific organization, once that is obtained you will ask the web service for the pertinent data
  - The API for the services is available at: <http://www.ist.rit.edu/api/>
  - All functionality is handled by your calls to appropriate services
  - The interface is responsive to the data
- It should work just like project 2, consuming the REST service. Display will be different.
- Professional graphics. No blah mono-color C# displays
- You will include the Map and Contact Form (displayed without errors)
- Show pictures of faculty/staff members equally sized and no loss of the image
- Include comments in the code to provide your name and clearly explain functionality
- Your name must be on the main Form title when it is first displayed, at end of IST header is good
- When done, zip up the entire project and turn it in ALONG with a ReadMe.txt file that includes anything you think is above and beyond the base requirements. What you describe will be considered for extra credit
- **DUE DATE: Thursday 12/7/2017 by 11:59pm – No office hours Monday.**

### Grading

---

If you complete all of the above, you will receive a B... A good project (worthy of a B) will have all required elements, implemented correctly and with attention to detail. An excellent project (worthy of an A) will have everything a B project has, plus something extra - evidence that you intend to excel. Perhaps it explores an area we didn't cover in detail in class, or would merit me recommending you to a client who needed similar work done on a .NET project. This needs to be specified as "Extras:" in your file named **README.TXT**.

Extra items:

- Level of complexity
- Thoroughness of interface
- Code abstraction
- DLLs beyond REST that was given in class

Avoid:

- Long text on one line that requiring right scrolling to read, or end of text is unreachable
- Combining text for items like Research. Don't list all research one after another. Allow which to choose
- Making popups to display everything requiring closing after each click
- Loading everything at the beginning, causing a long delay before the application starts without any indication to the user the program has been launched and is working on retrieving information

*PS - be careful - go for the "B" first and get it working... You will receive no credit for "extras" if basic functionality is missing.*

Grading Rubric:

Write functionality to obtain all the specific areas, the basic 'core' ones that show from the start to the on-demand ones that are requested.

Include comments in the code to clearly explain functionality

**Your name on UI title**

Zip, Readme, URL to your site

ABOUT

DEGREES (3)

MINORS (3)

EMPLOYMENT

MAP

PEOPLE

RESEARCH

RESOURCES

NEWS

COURSES

CONTACT FORM

Load API information when needed. No preloading at startup.

Display reasonable

What were the additions?

2nd DLL?

Do not reload information for a page on the 2<sup>nd</sup> visit