
Article

[Iryna Mykhailova](#) · Dec 22, 2021 6m read

Using IRIS at university (and a fun task)

As you can see in my profile I teach at a university and I wanted to share my take on teaching IRIS (or Caché before it).

It ' s been some time but I remember seeing a series of articles by [@Yuri Marx](#) earlier this year about the course " Developing with InterSystems Objects and SQL " . He actually gave a brief description of what ' s done [Day 1](#), [Day 2](#), and [Day 3](#) with comments from instructor [@Joel Solon](#). And I thought to myself, it may be interesting to share my experience.

Some history before I actually write about what I teach my student.

Right after I graduated, got my master ' s degree, and returned to work at the university, our department decided to update our programs and add several new courses to the general curriculum. One of those courses was " Postrelational databases " . It is taught to first-year masters. In the beginning, it consisted of 72 hours of lectures and 72 hours of practice. There were lots of changes and right now it ' s 72 hours of lectures and 36 hours of practice during the autumn semester.

Since I was a new hire and a young and bright one at that, I was given this new discipline. To say that I was surprised and horrified is to say nothing. First of all, I haven ' t had any practical experience in teaching at all. Secondly, I only had three summer months to learn a completely new to me technology and prepare a course. Luckily, it was decided for me which database should be taught. And this database turned out to be InterSystems Caché.

Anyways, I was more or less prepared and ready to roll, and then the questions from my dear students began. For example, why do they have to study this exact database, where and how can they use this database, etc. Since it was 2010, I was young and inexperienced, and there weren ' t a lot of books on the subject, I decided to go straight to the source, namely InterSystems. Somehow, I ended up talking to Mr. Solon and he gave me some great tips and also referred me to [@Evgeny Shvarov](#). And from there it all became much easier and clearer.

In the following years, I made quite a few changes to the contents of my course depending on the modern trends and demands. In the beginning, to show how to utilize this database from other applications I used Java binding and .NET Managed Provider. Later on eXTreme for Java appeared and then eXTreme for .NET. ZEN was taught and forgotten after a year or two. Now it ' s RESTful and SOAP services and CSP. A lot has changed, but the core rests practically the same.

Since we are talking master level here I have a lot of expectations. First and foremost, I expect my students to know the object-oriented paradigm and be able to draw correct UML class diagrams. Secondly, I expect them to know relational databases (including indexes, keys, etc) and SQL. Finally, they should have a working knowledge of at least simple web development.

Considering all the prerequisites, the following is the outline of my Syllabus.

Section1. Overview of postrelational databases: object-oriented, object-relational, and different kinds of NoSQL databases with descriptions of models and examples. Multimodel databases and examples.

Section 2. Architecture, structure, and basics of administration of IRIS as an example of postrelational DBMS. Working with databases and namespaces. Syntax, commands, variables, expressions, and some functions of ObjectScript. Also user code, exceptions, and transactions.

Section 3. Hierarchical model. Lists and multidimensional arrays (and globals). Functions to work with lists and arrays. And here comes the first task – design a global that has 4 subscript levels and use at least 4 functions to work with arrays to do something with the data.

Section 4. Object model. Classes, hierarchy, inheritance, parameters, different types of properties (and how to use them), different types of methods (using examples of calculated properties, Unit testing, creation of test data using Populate, user datatypes), parameters (and how to use them). How the data is stored and how to set up storage. Tasks here include designing a class diagram with classes that have different types of properties (references, built-in objects, both types of relationships, both types of collections, and a stream); creating these classes and several objects of each class, and linking everything together in IDE; creating a calculated property, a user datatype, Unit tests to check that all the constraints work (like min or max length/value, required property, unique property, etc) and generating test data.

Section 5. Relational model. Correlation between the object model and the relational model. Embedded (simple statements and cursors) and dynamic SQL. Class queries (both based on SQL and COS). Implicit join. IDKEY and accompanying methods. The task here is to add different types of SQL queries to classes from previous tasks with joins and parameters, create a trigger, change the unique property to IDKEY and see how it changes the related globals.

Section 6. Accessing data from outside IRIS. CSP and all about it. RESTful service and client (classes to work with JSON). SOAP service and client (classes to work with XML). The task here is to choose a class from previous tasks and create a CSP page that lists all the objects of the chosen class and gives the functionality to edit, create and delete objects. Then do the same things using RESTful and SOAP services (basically create methods to return all objects, one specific object, update and delete one specific object).

And that 's it. Apparently now, that there is an Embedded Python available, I will probably switch to it because all the questions and comments about the syntax of ObjectScript really get on my nerve.

Of course, the COVID-19 and lockdowns introduced some changes to the way how it all works. But IMHO those changes are for the best. And it really depends on how things are organized. For example, one of the advantages is that now I can actually show the students the usefulness of knowing all the stuff I talk about the whole semester.

In general, at the end of the semester, they have a control work that checks their knowledge of the main sections of the course. Before, they wrote it in class and I could supervise them. After, because the pandemic happened quite unexpectedly, I had to improvise and somehow provide each student with a unique set of tasks. Unfortunately, at that point, there was no software readily available so I decided to write a simple web portal to deal with this control work. Since it 's been almost two years now my simple portal turned into a RESTful service that randomly selects questions from different sets, assigns them to students, and receives the answers. All very modern and easy to use (and to write) 😊

As it is now the time for my current students to write this control work (and to have some fun here in this community) I dare you to take the test and send your answers J Of course the questions will be different (and easier) from the ones my students get. And I will give you the result J I 'm going to use the InterSystems Learning Labs server so you have only a couple of days to do the deed and check your knowledge.

The class diagram, that I use in the questions, is as follows:

To get the questions just send a GET request with your name to the following address:

<http://52773-1-e5a0b608.labs.learning.intersystems.com/community/task/<Yourname>>

You will get your set of 4 questions:

The screenshot shows a web browser with a single tab titled "http://52773-1-e5a0b608.l...". The address bar shows a GET request to `http://52773-1-e5a0b608.labs.learning.intersystems.com/community/task/Iryna Mykhailova`. Below the address bar are tabs for HEADERS, AUTHORIZATION (0), ACTIONS (0), CONFIG, and CODE SNIPPETS. The HEADERS tab is active, showing a "COPY" button and a "Text editor" toggle. Below this is a section to "Add a header to the HTTP request." with an "ADD" button. The RESPONSE tab is also visible, showing a status of "200 OK" and a "CLEAR" button. The response body is a JSON object:

```
1 { "Name": "Iryna Mykhailova", "Question1": "Difference between a neutron star and a black hole.", "Question2": "Using ObjectScript syntax create an instance of class Test and set all its properties", "Question3": "Using IRIS SQL insert a row with all the fields of table Test", "Question4": "On the page page1.csp in method OnPreHTTP there is code: set %response.Redirect = 'page2.csp'. What will be in the address bar after opening page1.csp?" }
```

To send answers just send a POST request to the following address:

<http://52773-1-e5a0b608.labs.learning.intersystems.com/community/answer/<Yourname>>

with a JSON in a body:

```
{  
  "Answer1": "answer 1",  
  "Answer2": "answer 2",  
  "Answer3": "answer 3",  
  "Answer4": "answer 4"  
}
```

The screenshot shows a REST client interface with a POST request to `http://52773-1-e5a0b608.labs.learning.intersystems.com/community/answer/Iryna Mykhailova`. The request body is a JSON object with four answer fields. The response is a 200 OK status with a response time of 344 ms and a size of 15 Bytes. The response body is a JSON object with a status field set to "OK".

```
Raw input
```

```
1 {
2   "Answer1": "answer1",
3   "Answer2": "answer2",
4   "Answer3": "answer3",
5   "Answer4": "answer4"
6 }
```

```
JSON
```

```
1 { "Status": "OK" }
```

200 OK Time: 344 ms Size: 15 Bytes

I will check the answers at some point (because there is this pesky thing called time difference) and you can get your results by sending GET request to the following address:

<http://52773-1-e5a0b608.labs.learning.intersystems.com/controlwork/marks/<Yourname>>

The screenshot shows a REST client interface with a GET request to `http://52773-1-e5a0b608.labs.learning.intersystems.com/community/marks/Iryna Mykhailova`. The response is a 200 OK status with a response time of 280 ms and a size of 53 Bytes. The response body is a JSON object with mark counts for each answer and an overall score.

```
GET
```

```
http://52773-1-e5a0b608.labs.learning.intersystems.com/community/marks/Iryna Mykhailova
```

COPY ☐ Text editor

Add a header to the HTTP request.

+ ADD

```
Response
```

```
1 { "Mark1": 2, "Mark2": 2, "Mark3": 0, "Mark4": 1, "Overall": 5 }
```

200 OK Time: 280 ms Size: 53 Bytes

If the marks are empty, it means I haven't yet had the time to check (or I'm not up yet). The points are as follows:

0 points mean that the answer is completely wrong.

1 point means that the answer is more or less correct.

2 points mean that the answer is correct.

The first person to send the answers will get a virtual hug from me (or a real one, if we meet in person)



Anyway, if you have any comments or questions about my course, don't hesitate to write them in the comments section.

[#Other](#)

Source URL: <https://community.intersystems.com/post/using-iris-university-and-fun-task>