Article
[Yuri Marx](#) · Mar 1, 2021 3m read

Day 4: Developing with InterSystems Objects and SQL

I'm participating in the Developing with InterSystems Objects and SQL with Joel Solon. The course is very nice and I will share with you some tips I got during the training. Tips presented in the day 4:

1. All data are stored in Globals and Global names start with ^. Example of global: ^animal. Global can have multiples locations to the data ("sub data"). Example: ^animal(1).
2. ^%* globals are accessible from any system-wide (from any namespace).
3. Globals enables IRIS to support multimodel data (object, relational, document, multidimensional, etc).
4. To see globals go the Management Portal > Explorer > Globals > Select Global > View or in Terminal type do ^%G or zwrite ^global.
5. There are an automatic correspondence between persistent classes and SQL Tables:
    1. Package is SQL Schema;
    2. Class is a Table;
    3. Property is a Column;
    4. Method is a Store Procedure; (when sqlProc);
    5. Relationship between classes is a SQL Foreign Key constraint (must be bi-directional);
    6. Object is a Row.
6. One table can correspond to multiple classes, but serial class is part of the table of the persistent class (don't have a specific table).
7. One class can correspond to multiple tables.
8. We have some classes types:
    1. Non-registered: not class object (container for methods only);
    2. Registered: transient objects;
    3. Persistent: SQL persistence in tables;
    4. Serial: SQL persistence in the main table (serial is embedded);
    5. Datatype: not class object is used to do new validations and conversions to base data types.
9. Classes can be composed by:
    1. Properties;
    2. Methods;
    3. Class queries: SQL Select statements;
    4. Parameters: user constants or system constants to configure the class behavior;
    5. Foreign keys: to referencial integrity;
    6. Indexes: to improve performance and do unique values;
    7. Triggers: fire methods associated with persistence events;
    8. XData: XML or JSON definitions associated with the class;
    9. Storage: description of the data storage.
10. Classes by convention has first letter of the word in Capital. Example: CountryOrigin. Parameters are all capital. Example: COLORNUMBER.
11. Class attributes qualify/configure a class. Example [SqlTableName = Animal] set the table name to a class. [Final] not allows inheritence. [Private] not allows call methods or use properties for non-subclasses.
12. Internally IRIS generate Get and Set to properties, and are not visible but can be declared to change the behavior.
13. Is possible override a method of the superclass, to do this, repeat the class name, the arguments. Is possible increase the number of arguments, not decrease.
14. Use ##super() to call base class method.
15. To create abstract class use [Abstract] and prevent instantiation.
16. Is possible extends multiple classes. Example Class Person extends (%Persistent, %Animal). (Persistent must be the fisrt in the extends, see joel tip in the comments)
17. REST is REpresentational State Transfer. Is based in the HTTP protocol. Use HTTP verbs: GET (select),

POST (insert), PUT (update) and DELETE (delete).
18. For expose your class as REST resource extend from %CSP.REST.
19. Use URLMap inside XData block to configure the routes of your REST service. Using the Portal, create a Web Application, enable REST and specify Dispatch class.
20. %JSONAdaptor provides convertion between objects and JSON. Use obj.%JSONImport(jsonObj) to assign DynamicObject to a object. Use obj.%JSONExportToString(.jsonString) to write a JSON String to a object.
21. %JSON.Formatter format a JSON String for human readability.

#Globals #Object Data Model #REST API #SQL #InterSystems IRIS