



You completed this test on 2020/03/19, 20:23

Your score is 71.43%

CORRECT

Which of following uses the same naming convention that Kentico uses?

FooObject, FooObjectProvider

FooClass, FooClassProvider

FooInfo, FooInfoProvider

FooData, FooDataProvider

CORRECT

What can be a result of calling `UserInfoProvider.SetUserInfo()` method?

(multiple answers)

Created user

Updated user

Deleted user

Information logged into Event log application about created / updated user

CORRECT

What are the benefits of ObjectQuery?

(multiple answers)

- Strongly typed and enumerable results
 - Protection against SQL injection if parametrized conditions are used
 - Lazy loading of results
 - Syntax is independent of SQL version
-

CORRECT

If related data you need to retrieve is stored in two tables, which of these approaches is best with regards to performance?

Retrieve data from table A, loop through each item and get related data from table B.

Retrieve data from table A and B and join them in memory.

- Use object query to join two objects and retrieve data in one SQL call.
-

CORRECT

Which of the following method(s) would improve performance when retrieving only the user's first and last name?

`UserInfoProvider.GetUsers().GetColumns("FirstName", "LastName");`

- `UserInfoProvider.GetUsers().Columns("FirstName", "LastName");`

`UserInfoProvider.GetUsers().Column("FirstName").Column("LastName");`

`UserInfoProvider.GetUsers();`

WRONG

Which of the following is true?

(multiple answers)

- You can use `CacheHelper.Cache()` method to cache any object.

- Some InfoProviders can store the returned object in a hashtable when calling its Get<Name>Info() method, so custom caching is not needed.
 - You can access cached objects in the application memory via Kentico debugging tools.
-

WRONG

Select the correct answer(s) about custom module API:

(multiple answers)

- The <object>Info and <object>InfoProvider are generated and saved to the file system automatically when the module class is created.
- The <object>Info and <object>InfoProvider can't be modified in the Kentico admin interface.
- Kentico generates the reference fields for objects automatically in the <object>Info class, but not on the DB level as foreign key.

Strongly typed API is only available for custom modules, not custom tables

Passed. Let's continue.