

# EXAMUNION

Certified IT Exam Material Authority



Accurate study guides, High passing rate!  
We offer free update service for one year!

<http://www.examunion.com>

**Exam** : **70-482**

**Title** : Advanced Windows Store  
App Development using  
HTML5 and JavaScript

**Version** : Demo

## Topic 1, Scenario 1

### Application Information

You are developing two Windows Store apps by using JavaScript: a Personal Trainer app and a Client app. The apps will allow personal fitness trainers to interact with their remote clients.

#### Business Requirements- Personal Trainer Application

The Personal Trainer app must allow trainers to perform the following tasks:

- Create and store video and audio recordings of workout routines.
- View the profile and workout recordings for only one client at any time.

#### Business Requirements- Client Application

The Client app must allow clients to perform the following tasks:

- Browse a list of the trainer's workout recordings.
- Record workouts by using the built-in webcam.
- Play, pause, restart, and stop workout recordings.
- If the capability is supported, allow the client's webcam to pan as the client moves around the room.
- Upload workout recordings for trainer review.
- Update their individual profiles to indicate workouts completed, calories burned, and current weight.

The Client app must validate that the client's subscription is valid.

#### Technical Requirements- General

The Personal Trainer and Client apps must meet the following technical requirements:

- Connect to the Internet.
- Store workout recordings in the cloud.
- Enable retrieval of workout recordings by using a custom URL.
- Encapsulate the video player in a custom control.
- Identify the maximum zoom of the user's webcam in millimeters.
- Store client profiles in XML files in the trainers' Documents folders to allow for disconnected editing.
- Synchronize the XML files with cloud storage by using a background task when the Internet is available.
- Send trainer workout videos to cloud storage by using a background task when the trainer's device is idle. Indicate the status of the upload operation each time the trainer starts the app. Suspend the background task when the Internet is not available.
- Separate business and complex logic into WinMD components. The solution debugging settings must include the WinMD components.

#### Technical Requirements- Hardware Requirements

The Personal Trainer and Client apps must support the following hardware requirements:

- Windows 8
- Webcam, microphone, and speakers
- Internet connection

While testing the apps, you identify the following issues:

- When you start the app for the first time, the system displays this warning message: "This app needs permission to use your camera, which you can change in the app's settings."
- When you run the loadClientProfile ( ) method in the clientData.js file, you receive an "Access Denied" exception.
- The findCamera ( ) method in the video.js file throws an exception on some devices.
- The recordVideo ( ) method in the video.js file throws an exception when the device does not support

tilting.

### **clientData.js**

```
CD01 function loadClientProfile() {
CD02   var fop = new Windows.Storage.Pickers.FileOpenPicker();
CD03   fop.viewMode = Windows.Storage.Pickers.PickerViewMode.thumbnail;
CD04
CD05   fop.fileTypeFilter.replaceAll([".xml"]);
CD06
CD07   (function (file) {
CD08     if (file) {
CD09       display(file);
CD10     }
CD11     else {
CD12       processError(file);
CD13     }
CD14   });
CD15 }
CD16
CD17 function saveClientProfile() {
CD18   var sp = new Windows.Storage.Pickers.FileSavePicker();
CD19   sp.defaultFileExtension = ".xml";
CD20   sp.suggestedFileName = "New Client";
CD21
CD22
CD23   sp.pickSaveFileAsync().then(
CD24     function (file) {
CD25       if (file) {
CD26         displaySaved(file);
CD27       }
CD28       else {
CD29         processError(file);
CD30       }
CD31     });
CD32 }
```

**video.js**

```
VD01 function recordVideo() {
VD02   var device = new Windows.Media.Capture.MediaCapture();
VD03   var videoDev = device.videoDeviceController;
VD04   var canTilt = videoDev.tilt.capabilities.supported;
VD05
VD06
VD07   ...
VD08 }
VD09
VD10 var cameraID;
VD11
VD12 function findCamera() {
VD13   var deviceInfo = Windows.Devices.Enumeration.DeviceInformation;
VD14   deviceInfo.findAllAsync
(Windows.Devices.Enumeration.DeviceClass.videoCapture).then (function (devices)
{
VD15     cameraID = devices[0].id;
VD16   }, errorHandler);
VD17 }
VD18
```

**background.js**

```
BG01 function registerBackgroundTask(condition) {
BG02   var builder
= new Windows.ApplicationModel.Background.BackgroundTaskBuilder();
BG03   builder.name = "videoLoader";
BG04   builder.taskEntryPoint = "background.js";
BG05   builder.setTrigger(
BG06     Windows.ApplicationModel.Background.SystemTrigger(
BG07
BG08   ));
BG09
BG10   ...
BG11 }
BG12
BG13 function unregisterBackgroundTask() {
BG14
BG15   var i = tasks.hasCurrent;
BG16   while (i) {
BG17     var task = tasks.current.value;
BG18     if (task.name === "videoLoader") {
BG19       task.unregister(true);
BG20     }
BG21     i = tasks.moveNext();
BG22   }
BG23 }
```

1.You need to complete the code to start the background task.

Which code segment should you insert at line BG07?

- A. `Windows.ApplicationModel.Background.SystemTriggerType.connectedStateChange, true`
- B. `Windows.ApplicationModel.Background.SystemTriggerType.networkStateChange, false`
- C. `Windows.ApplicationModel.Background.SystemTriggerType.sessionConnected, true`
- D. `Windows.ApplicationModel.Background.SystemTriggerType.internetAvailable, false`

**Answer: D**

2.You need to identify the required camera specifications.

Which code segment should you insert at line VD06?

- A. `varmaxZoom = videoDev.zoom.capabilities.max;`
- B. `varcameraType = videoDev.extendedZoomProperties("cameraType");`
- C. `varcameraZoom = videoDev.zoom;`
- D. `varminZoom = mediaCaptureSettings.min.millimeters;`

**Answer: A**

3.You need to attach the background task.

Which code segment should you insert at line BG09?

- A. `var task = builder.register ( );`
- B. `var task = Windows. ApplicationModel.Background.BackgroundTaskBuilder.insert (builder);`
- C. `var task: = Windows.ApplicationModel.Background.BackgroundTaskBuilder.insert (builder, this);`
- D. `var task = builder.setTrigger ( );`

**Answer: A**

4.You need to set the default storage location for the client profiles.

Which code segment should you insert at line CD04?

- A. `fop.defaultFolder = Windows.Storage.Pickers.PickerLocationId.documentsLibrary;`
- B. `fop.defaultFolder = environment.getFolderPath (environment.specialFolder.applicationData);`
- C. `fop.suggestedStartLocation = "%AppData%";`
- D. `fop.suggestedStartLocation =Windows.Storage.Pickers.PickerLocationId.documentsLibrary;`

**Answer: A**

5.You need to prevent the exception that is being thrown by the `findCamera( )` method.

What should you do?

- A. Check the devices collection for multiple devices.
- B. Place a try block immediately after line VD12 and a catch block immediately before line VD17. In the catch block, display the message property of the exception object to the user.
- C. In line VD10, set the `cameraID` variable to null.
- D. Check the devices collection for null before setting the `cameraID` variable.

**Answer: D**