



Xamarin Evolve 2014

Working with Cloud Services

Microsoft Azure, Parse, Dropbox

Craig Dunn
craig@xamarin.com

 Xamarin University

No phone is an island

Xamarin University

Agenda

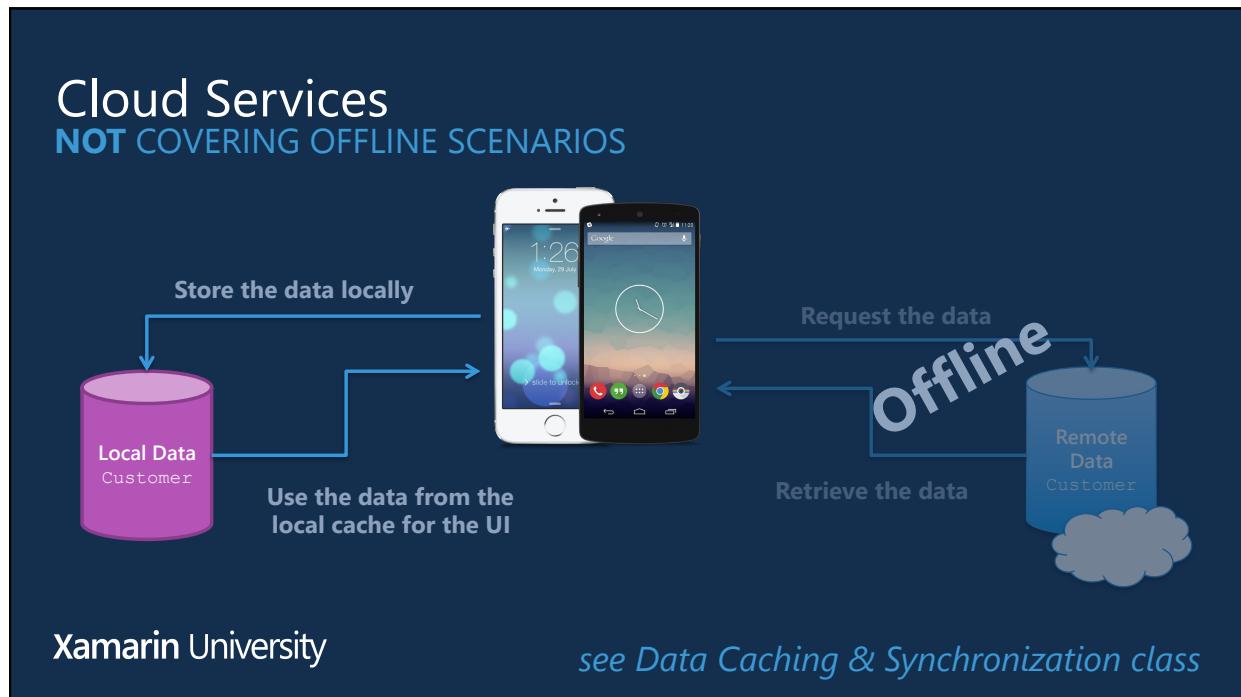
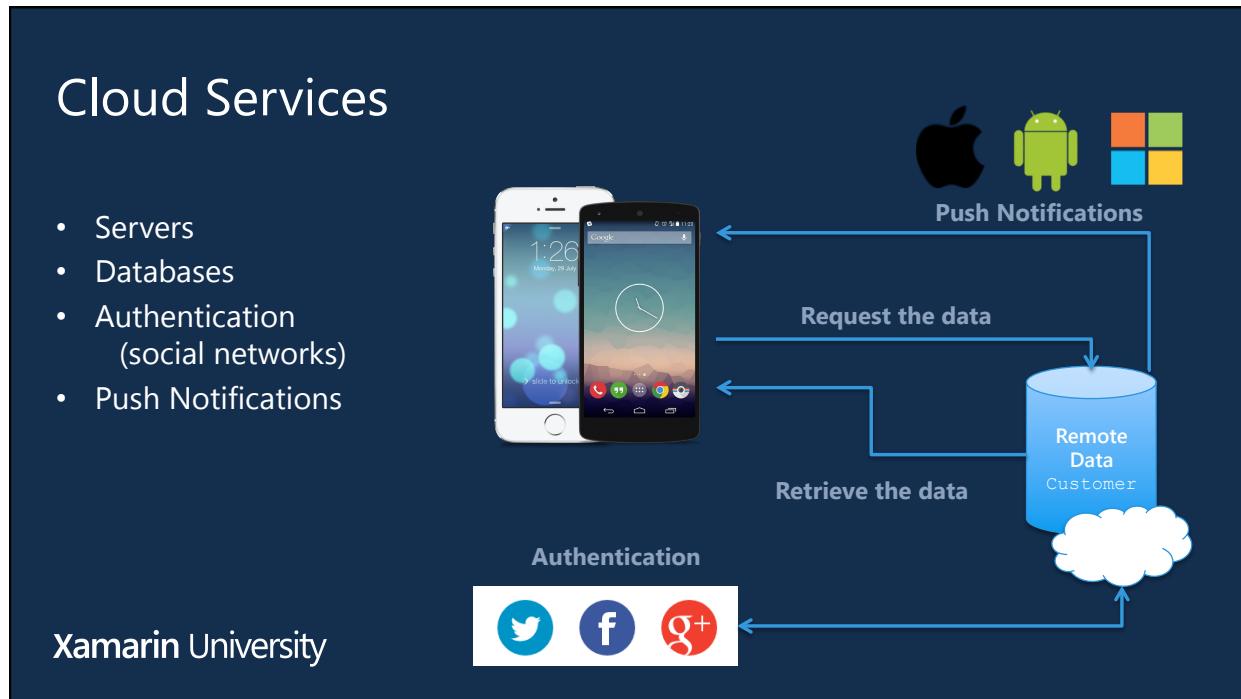
1. What is a cloud?
2. Microsoft Azure
3. Parse
4. DropBox

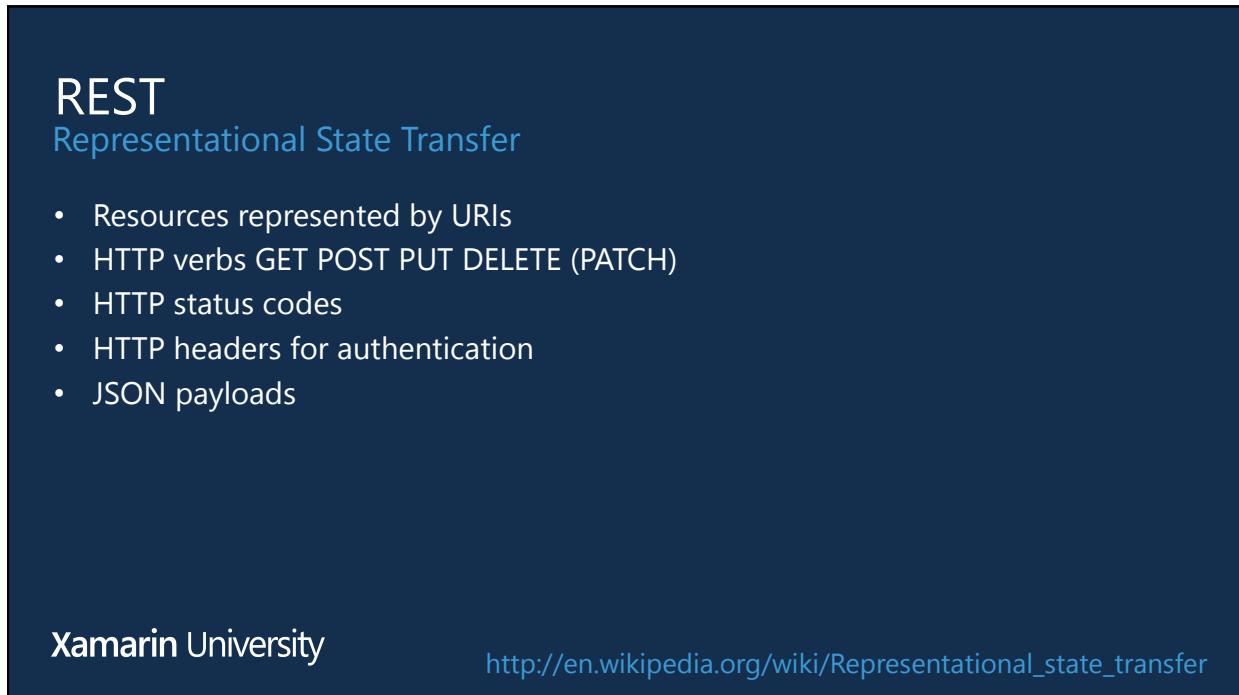
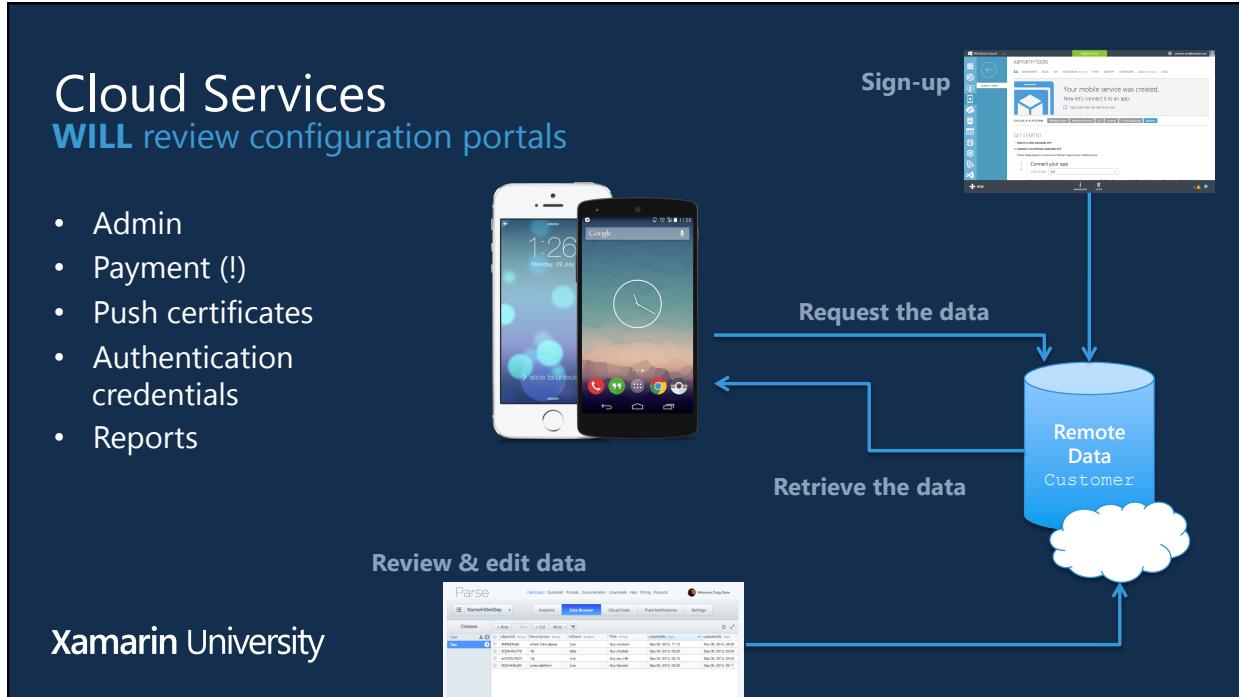
Xamarin University



What is a cloud?

Xamarin University





REST APIs – Azure

Representational State Transfer

```
string UpdateUrl = "https://"+subdomain+".azure-mobile.net/tables/TodoItem/{0}";  
public static void UpdateTodo(TodoItem t)  
{  
    WebClient client = new WebClient();  
    try {  
        client.Headers.Add (HttpRequestHeader.Accept, "application/json");  
        client.Headers.Add (HttpRequestHeader.ContentType, "application/json");  
        client.Headers.Add ("X-ZUMO-APPLICATION", MobileServiceAppId);  
        var payload = t.ToJson ();  
        var response = client.UploadString (String.Format (UpdateUrl,t.Id), "PATCH", payload)  
        // ...and wait...  
        var responseString = response;  
        var responseJson = JsonValue.Parse (responseString);  
    } catch (System.Net.WebException e) {  
        Console.WriteLine ("X-ZUMO-APPLICATION update failed" + e.Message);  
    }  
}
```

Xamarin University

<http://msdn.microsoft.com/en-us/library/azure/jj710108.aspx>

REST APIs – Parse

Representational State Transfer

URL	HTTP Verb	Functionality
/1/classes/<className>	POST	Creating Objects
/1/classes/<className>/<objectId>	GET	Retrieving Objects
/1/classes/<className>/<objectId>	PUT	Updating Objects
/1/classes/<className>	GET	Queries
/1/classes/<className>/<objectId>	DELETE	Deleting Objects

"X-Parse-Application-Id: 60oS1RVeH1iFt4m1eDDyorCDDiGyIM5BevmTztZ9
"X-Parse-REST-API-Key: m06f6mzIKrkRnXpKZAUpWb5rZtqC5ixGaDeFFYe"

Xamarin University

<https://parse.com/docs/rest>

REST APIs – Dropbox

Representational State Transfer

- Request

```
https://api.dropbox.com/1/datastores/get_snapshot?handle=<DATASTORE-HANDLE>
Authorization: Bearer <YOUR-ACCESS-TOKEN>
```

- Response

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"rows": [], "rev": 0}
```

Xamarin University

<https://www.dropbox.com/developers/datastore/tutorial/http>

Client Libraries

Best or worst idea ever?

- Azure

```
// Initialize with your URL and Key
Client = new MobileServiceClient ("https://xamarin-todo-cd2.azure-mobile.net/",
"APPLICATION_KEY", this);
// Work with a table of data
todoTable = Client.GetTable<TodoItem>();
// Insert a new item
await todoTable.InsertAsync(todoItem);
```

- Parse

```
// Initialize with your URL and Key
ParseClient.Initialize ("APP_ID", "WINDOWS_KEY");
// Save changes
await ToParseObject(todoItem).SaveAsync();
```

Xamarin University

Dynamic Schema

Best or worst idea ever?

```
public class TodoItem
{
    public TodoItem () {}

    public string ID { get; set; }

    [JsonProperty(PropertyName = "text")]
    public string Name { get; set; }

    [JsonProperty(PropertyName = "notes")]
    public string Notes { get; set; }

    [JsonProperty(PropertyName = "complete")]
    public bool Done { get; set; }

    public string TempField101_blah { get; set; }
}
```

Xamarin University

Implementing Cloud Services

- Portal
- Client Library
- Code

Xamarin University

Microsoft Azure

Xamarin University

Microsoft Azure

- Mobile Services (with database)
- Push notifications
- Authentication
- Server-side programming
- *many many more!*

Xamarin University



Portal
windowsazure.com

xamarin-todo

DASHBOARD DATA API SCHEDULER PREVIEW PUSH IDENTITY CONFIGURE SCALE PREV

Your mobile service was created.

Now let's connect it to an app.

CHOOSE A PLATFORM Windows Store Windows Phone 8 iOS Android HTML/JavaScript Xama

GET STARTED

CREATE A NEW XAMARIN APP

CONNECT AN EXISTING XAMARIN APP

Component
...or use NuGet

Add Packages

nuget.org

azur... mobile

<input checked="" type="checkbox"/>	 Windows Azure Mobile Services	54,692
<input type="checkbox"/>	 Microsoft Azure Mobile Services .NET Backend Resource Broker	150
<input type="checkbox"/>	 Microsoft Azure Mobile Services .NET Backend Azure Storage	925
<input type="checkbox"/>	 Azure Mobile Services for Windows Phone	582
<input type="checkbox"/>	 Windows Azure Mobile Services for WinJS	4,367

Show pre-release packages Close Add Package

Azure Mobile Services 1.0.4

Microsoft

15 ratings Compatible with

Store data in the cloud, authenticate users, and send push notifications.

Getting Started License Website

Download

Windows Azure Mobile Services 1.0.4

Publisher Microsoft

Category Cloud Services

Price Free

This client library enables client applications to connect to Windows Azure Mobile Services. Mobile Services allows you to develop an app with a scalable and secure backend hosted in Windows Azure. You can incorporate structured storage, user authentication and push notifications in minutes. For more information, check out <https://www.windowsazure.com/mobile>.

Id WindowsAzure.MobileServices

Author Microsoft

Published 10/09/2014

Downloads 54,692

License View License

Project Page Visit Page

Dependencies Newtonsoft.Json (>= 6.0.4)
Microsoft.Bcl (>= 1.1.9)
Microsoft.Net.Http (>= 2.2.28)

Code

```
// Initialize with your URL and Key
Client = new MobileServiceClient ("https://.azure-mobile.net/",
"APPLICATION_KEY", this);
todoTable = Client.GetTable<TodoItem>();

// Work with a table of data
Items = await todoTable.ToListAsync();
Items = await todoTable.Where
    (todoItem => todoItem.Done == false).ToListAsync();

// Insert a new item
await todoTable.InsertAsync(todoItem);
```

Xamarin University

Validation Using Script

script

```
function insert(item, user, request)
{
    if (item.text.length < 5)
    {
        request.respond(statusCodes.BAD_REQUEST,
            'Surely you meant to type more than 5 chars?');
    }
    else
    {
        request.execute();
    }
}
```

c# code

```
catch (MobileServiceInvalidOperationException e) {
    if (e.Response.StatusCode == System.Net.HttpStatusCode.BadRequest) {
        CreateAndShowDialog (e.Message, "Validation error");
    } else {
        // another error that we are not expecting
        CreateAndShowDialog (e.Response.Content.ToString (), "Error");
    }
}
```

Xamarin University

Authentication Using Social Network Logins

configure

google settings

CLIENT ID 196205644313-5555555555559qaibnsctSqlr8spul.
CLIENT SECRET iuZSDP5FY55555555

set permissions

INSERT PERMISSION Only Authenticated Users

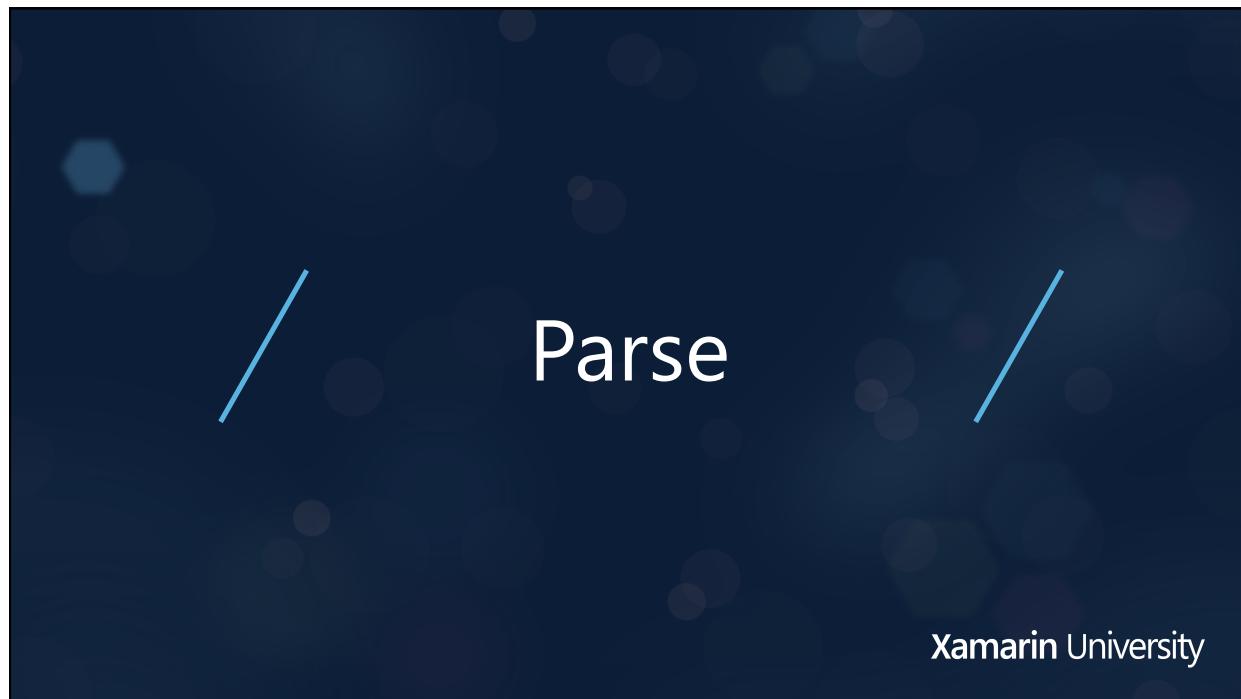
tag rows with user

```
function insert(item, user, request)
{
    item.userId = user.userId;
    request.execute();
}
```

filter rows by user

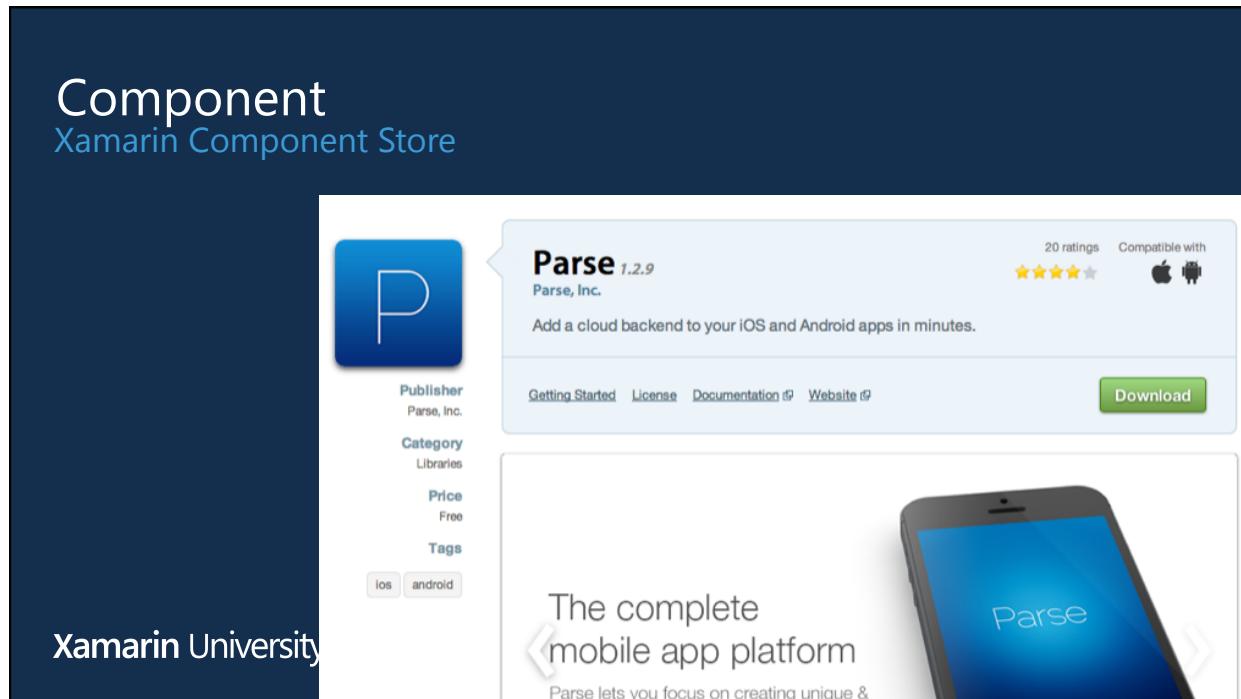
```
function read(query, user, request)
{
    query.where({userId : user.userId});
    request.execute();
}
```

Xamarin University



A screenshot of the Parse Data Browser interface. The header includes the Parse logo, a navigation bar with links to Dashboard, Quickstart, Tutorials, Documentation, Downloads, Help, Pricing, Products, and a user profile for "Craig Dunn". The main area shows a table of data with columns: Classes, objectId, String, Description, String, IsDone, Boolean, Title, String, createdAt, Date, updatedAt, Date. There are two entries under the "Task" class: one for "wheat thins please" and another for "Buy crackers". Both entries have their "IsDone" status set to "true".

Classes	objectId	Description	IsDone	Title	createdAt	updatedAt
User	9Mf630Rqfd	wheat thins please	true	Buy crackers	Sep 05, 2013, 17:12	Nov 06, 2013, 04:51
Task	2CDAiIRuFy6	1lb	false	Buy cheddar	Sep 04, 2013, 05:20	Sep 05, 2013, 23:00
	anCROURtO7	1qt	true	Buy soy milk	Sep 04, 2013, 05:15	Sep 05, 2013, 23:00
	XGDvN4Ea2X	cross-platform!	true	Buy Xamarin	Sep 04, 2013, 05:02	Sep 05, 2013, 23:00



Code

```
// Initialize  
ParseClient.Initialize (Constants.AppKey, Constants.WindowsKey);  
  
// get a list of data  
var query = ParseObject.GetQuery ("Task").OrderBy ("Title");  
  
// insert a new object  
await ToParseObject(todoItem).SaveAsync();
```

Xamarin University

Dropbox

Xamarin University

Dropbox

- Authentication
- Data Store
- Dropbox is DIFFERENT!



Xamarin University

Component
Xamarin Component Store

Dropbox Sync & Datastore 3.0.2

Publisher: Xamarin, Inc.

Category: Cloud Services

Price: Free

Published On: August 08, 2014 (47 days ago)

Tags: ios, android

30 ratings Compatible with:

Give your app its own private Dropbox client and leave the syncing to Dropbox.

[Getting Started](#) [License](#) [Website](#)

Installed

The Dropbox Sync API allows you to give your app its own private Dropbox client and leave the syncing to Dropbox.

- Focus on your data. The Sync API handles all the caching, retrying, and file change notifications.
- Writes are local so changes are immediate. The Sync API syncs to Dropbox behind the scenes.
- Your app works great even when offline and automatically syncs when it's back online.

Datastore API

Keep your app's structured data in sync with Dropbox. These days, your app need to store and sync more than just files. With the Datastore API, structured data like contacts, to-do items, and game state can be synced effortlessly. Datastores work across platforms, offline, and even support automatic conflict resolution.

Authenticating with Dropbox

Add the following lines of code to link a user's Dropbox account to your app:

Portal
Dropbox.com/Developers

Your apps

Create app

The Dropbox Platform gives you effortless access to hundreds of millions of Dropboxes. Whether you're building a server-based app with the Core API, syncing structured data with the Datastore API, or just grabbing files with the Chooser, you'll need to create an app to integrate with Dropbox.

TaskyDrop

Datastores		Records	
▼ default	monkeys	id	4etfRhPc8MbmlsXXFjrw
	tasks	Description	getting closer
		IsDone	False
		Title	Write TODO sample...
		id	53DzbNzRWaReZ4FKh3g2tw
		Description	Falling
		IsDone	False
		Title	Make monkeys dance!
		id	6R2TMBh4pxXn-1KsDwjqpEw

Code

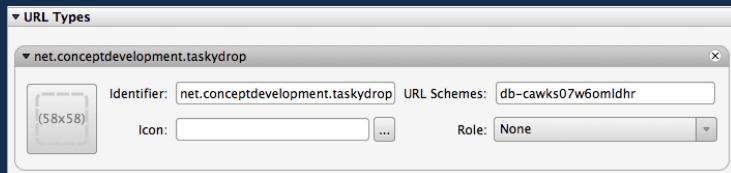
```
// Initialize
var manager = new DropBoxSync.iOS.DBAccountManager
    (DropboxSyncKey, DropboxSyncSecret);
DropBoxSync.iOS.DBAccountManager.SharedManager = manager;

// insert or save an object
var table = store.GetTable(tableName);
var record = table.GetRecord(todoItem.ID, out error);
if (record == null)
    table.Insert(todoItem.ToDictionary());
else
    record.Update(todoItem);
```

Xamarin University

Config

Auth redirects



```
public override bool OpenUrl (UIApplication application, NSURL url, string
sourceApplication, NSObject annotation)
{
    var account = DBAccountManager.SharedManager.HandleOpenURL (url);
    System.Threading.Tasks.Task.Factory.StartNew (() => {
        Init ();
    });
    return account != null;
}
```

Xamarin University

Summary

- It's ridiculously easy to add cloud services to Xamarin apps
- Choose Azure, Parse, Dropbox, or a host of other providers

Xamarin University

What's Next?

- Classes
 - Mobile Data
 - Web Services
 - Data Caching & Synchronization
 - Mobile Security
- Sessions
 - Building Cross-Platform Business Apps with Azure
 - Cross-platform syncing with Dropbox
- Mini-hacks!

Xamarin University

The slide features a dark blue background with a large, semi-transparent circular graphic on the left containing a white square icon. In the top right corner, the text "Xamarin Evolve 2014" is displayed. The main title "Working with Cloud Services" is centered in large, white, sans-serif font. Below it, the subtitle "Microsoft Azure, Parse, Dropbox" is shown in a smaller, white, sans-serif font. On the left side, the name "Craig Dunn" and email address "craig@xamarin.com" are listed in white. On the right side, the "Xamarin University" logo is present, consisting of a hexagonal icon with a stylized 'X' and the text "Xamarin University".

Xamarin Evolve 2014

Working with
Cloud Services

Microsoft Azure, Parse, Dropbox

Craig Dunn
craig@xamarin.com

Xamarin University