BASTA!
.NET, WINDOWS, VISUAL STUDIO

C# for Android and iOS

Xamarin

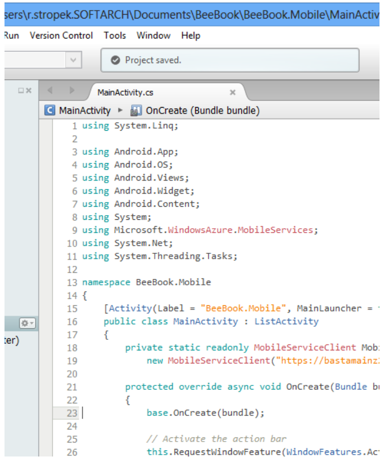
Inhalt

Sie sind erfahrener C#-Entwickler, die Welt von Android und iOS reizt Sie aber trotzdem? In dieser Session zeigt Ihnen Rainer Stropek, wie Sie mit den **Xamarin-Tools** Ihr C#-Wissen auf diese mobilen Plattformen mitnehmen können. Rainer stellt Ihnen die Tools vor und demonstriert an einem durchgängigen **Beispiel**, wie **plattformübergreifende C#-Codewiederverwendung** funktionieren kann.

Agenda

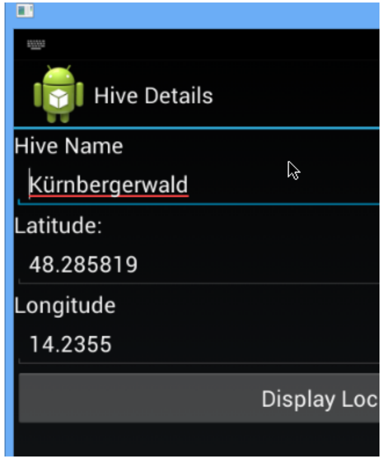
Xamarin

ate iOS, Android, M
d Windows apps in
Join our community of 406,499 devel


sers\r.stropek.SOFTARCH\Documents\BeeBook\BeeBook.Mobile\MainActiv

Run Version Control Tools Window Help
Project saved.
MainActivity.cs
MainActivity ▸ OnCreate (Bundle bundle)

```
using System.Linq;

using Android.App;
using Android.OS;
using Android.Views;
using Android.Widget;
using Android.Content;
using System;
using Microsoft.WindowsAzure.MobileServices;
using System.Net;
using System.Threading.Tasks;

namespace BeeBook.Mobile
{
    [Activity(Label = "BeeBook.Mobile", MainLauncher = 
    public class MainActivity : ListActivity
    {
        private static readonly MobileServiceClient Mob:
            new MobileServiceClient("https://bastamainz:

        protected override async void OnCreate(Bundle b
        {
            base.OnCreate(bundle);

            // Activate the action bar
            this.RequestWindowFeature(WindowFeatures.Ac
```


Hive Details

Hive Name
Kürnbergerwald
Latitude:
48.285819
Longitude
14.2355

Xamarin
Introduction

Bildquelle:
http://www.xamarin.com

Develop
Xamarin Studio
Visual Studio
Debugging

Bildquelle:
Screenshot Xamarin Studio

Example
App dev basics
Native APIs
Azure component
Code sharing

Summary
Key takeaways

Bildquelle:
http://www.flickr.com/photos/caveman
_92223/3347745000/

Introduction

What‘s Xamarin and what problems does it solve?

What Problem Does Xamarin Address?

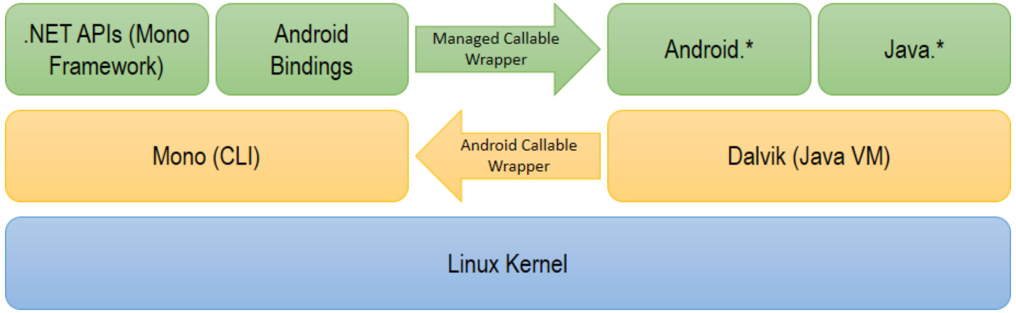
- Need to support a broad range of mobile devices
  Different platforms – Android, iOS, Windows Phone
  Different devices – smartphones, tablets
- Existing C# knowledge and experience
  Skilled C#/.NET developers and existing C#/.NET codebase
- Lack of knowledge about native development
  Java, Objective-C
- Can we cover mobile device market with our existing knowledge and tools?

Potential Solutions

- Build native apps
  Requires knowledge about C#, Java, and/or Objective-C
  Requires knowledge about the target platform
- Build mobile web sites
  Does it feel like a real native app?
  Lack of possibilities to fully use the underlying platform?
- Use a cross-platform development tookit
  E.g. Phonegap
- Xamarin: Existing tools & knowledge with bridge to native APIs

What is Xamarin?

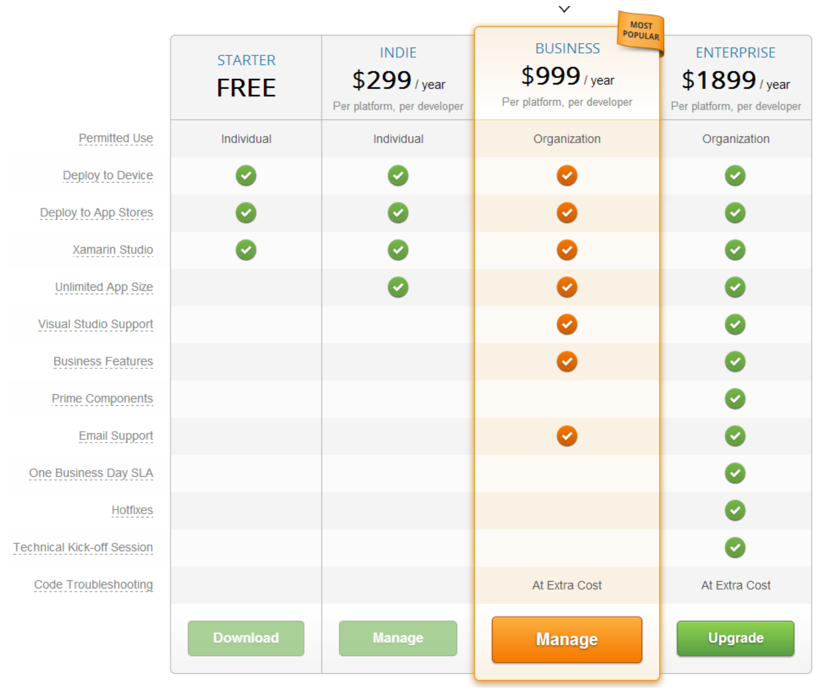
- Company founded by the initiators of the Mono project
- C# + Runtime + .NET BCL
  C# Compiler
  Implementation of the Common Language Infrastructure for Linux-based systems
  .NET Base Class Library
- Bridges to native API
  Callable Wrappers
- Development environments
  Xamarin Studio
  Visual Studio integration
  Component store



Pricing

- Xamarin is only free for very small apps
- Pricing per year and per developer

- Prices on the right as per Sept. 24th 2013
- For up-to-date prices see [Xamarin Store]()



	STARTER FREE	INDIE $299 / year Per platform, per developer	BUSINESS $999 / year Per platform, per developer MOST POPULAR	ENTERPRISE $1899 / year Per platform, per developer
Permitted Use	Individual	Individual	Organization	Organization
Deploy to Device	✓	✓	✓	✓
Deploy to App Stores	✓	✓	✓	✓
Xamarin Studio	✓	✓	✓	✓
Unlimited App Size		✓	✓	✓
Visual Studio Support			✓	✓
Business Features			✓	✓
Prime Components				✓
Email Support			✓	✓
One Business Day SLA				✓
Hotfixes				✓
Technical Kick-off Session				✓
Code Troubleshooting			At Extra Cost	At Extra Cost
	Download	Manage	Manage	Upgrade

Bridges

Java Example

```
public class ContactListCursorAdapter extends BaseAdapter {
   private Context mContext;
   private List<ContactEntry> mItems = new ArrayList<ContactEntry>();

public ContactListCursorAdapter(
   Context context, ArrayList<ContactEntry> items) {
      mContext = context;
      mItems = items;
}

public int getCount() {
      return mItems .size();
}

public Object getItem(int position) {
      return mItems .get(position);
}

public long getItemId(int position) {
      return position;
}

public View getView(int position, View convertView, ViewGroup parent) {
…
}
}
```

```
public class HiveAdapter : BaseAdapter<Hive>
{
   private IReadOnlyList<Hive> items = new List<Hive>();

   public HiveAdapter(Context context) {
      …
   }

   public override int Count { get { return this.items.Count; } }

   public override Hive this[int position] {
      get { return this.items[position]; } }

   public override long GetItemId(int position) { return
      this.items[position].Id; }

   public override View GetView(
      int position, View convertView, ViewGroup parent)
   {
      …
   }
}
```

Bridges

C# Example

getCount becomes *Count* property

getItem becomes indexer

getItemId becomes *GetItemId*

getView becomes *GetView*

Bridges

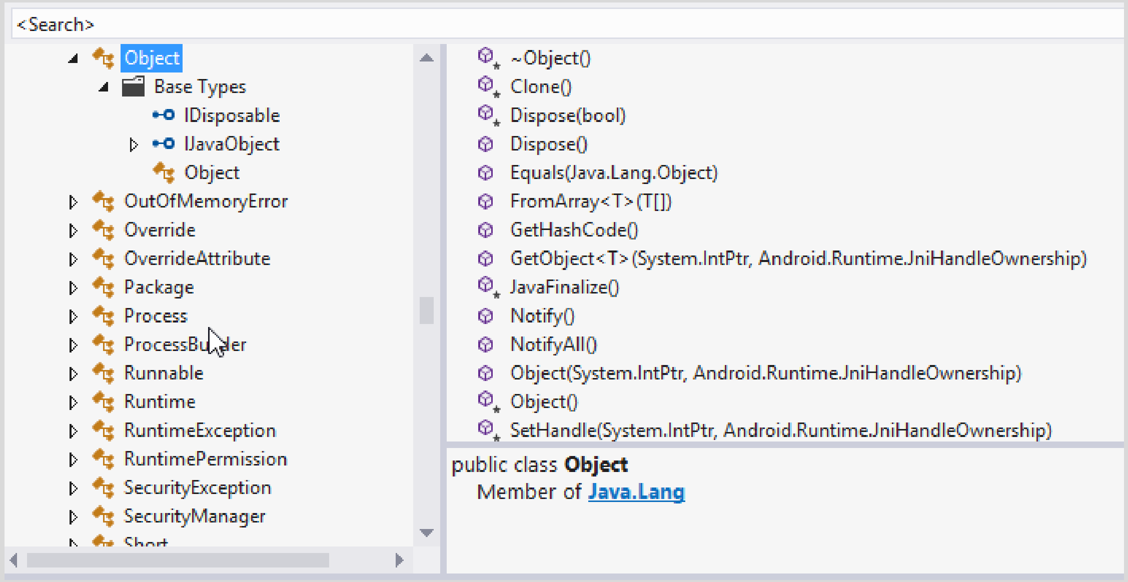
Events

Java Listener Interfaces
become *C# events*

```
// JAVA ===============================================================
someTextView.setOnClickListener(new View.OnClickListener() {
    public void onClick() {
        //Do Stuff
    }
});


// C# =================================================================
itemTextView.Click += (o, e) => {
  // Do Stuff
};
```

Wrapper Lifetime

IDisposable

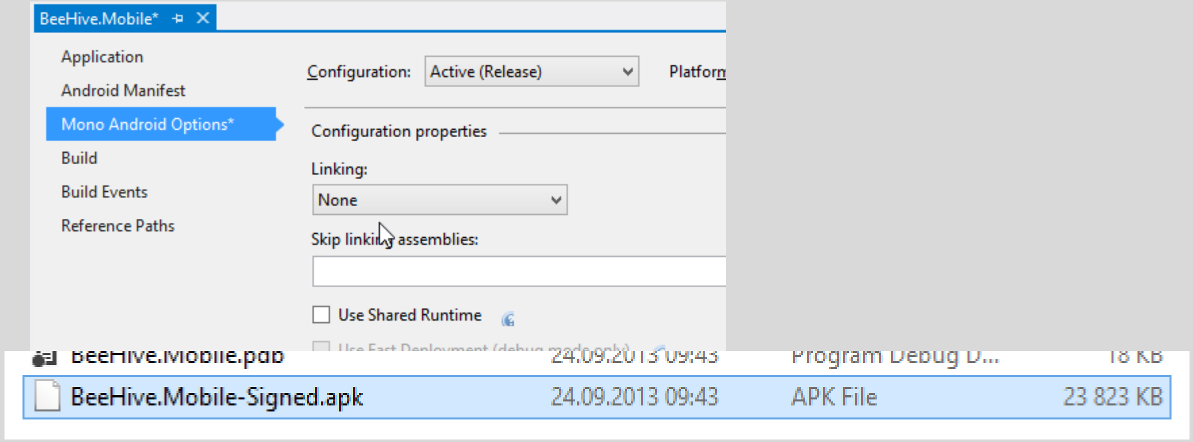


```
<Search>
◢ Object
   ◢ Base Types
        IDisposable
      ▷ IJavaObject
        Object
▷ OutOfMemoryError
▷ Override
▷ OverrideAttribute
▷ Package
▷ Process
▷ ProcessBuilder
▷ Runnable
▷ Runtime
▷ RuntimeException
▷ RuntimePermission
▷ SecurityException
▷ SecurityManager
▷ Short
```

```
~Object()
Clone()
Dispose(bool)
Dispose()
Equals(Java.Lang.Object)
FromArray<T>(T[])
GetHashCode()
GetObject<T>(System.IntPtr, Android.Runtime.JniHandleOwnership)
JavaFinalize()
Notify()
NotifyAll()
Object(System.IntPtr, Android.Runtime.JniHandleOwnership)
Object()
SetHandle(System.IntPtr, Android.Runtime.JniHandleOwnership)
```

public class **Object**
Member of Java.Lang

Note that all classes derived from *Java.Lang.Object* implement *IDisposable*


BeeHive.Mobile*

Application
Android Manifest
Mono Android Options*
Build
Build Events
Reference Paths

Configuration: Active (Release)

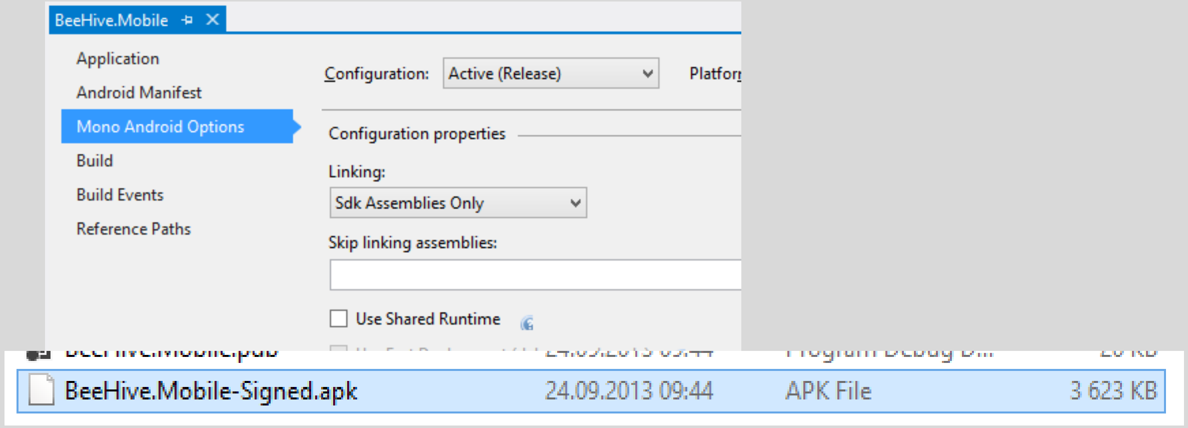
Configuration properties

Linking: None

Skip linking assemblies:

☐ Use Shared Runtime

```
BeeHive.Mobile-Signed.apk    24.09.2013 09:43    APK File    23 823 KB
```


BeeHive.Mobile

Application
Android Manifest
Mono Android Options
Build
Build Events
Reference Paths

Configuration: Active (Release)

Configuration properties

Linking: Sdk Assemblies Only

Skip linking assemblies:

☐ Use Shared Runtime

```
BeeHive.Mobile-Signed.apk    24.09.2013 09:44    APK File    3 623 KB
```

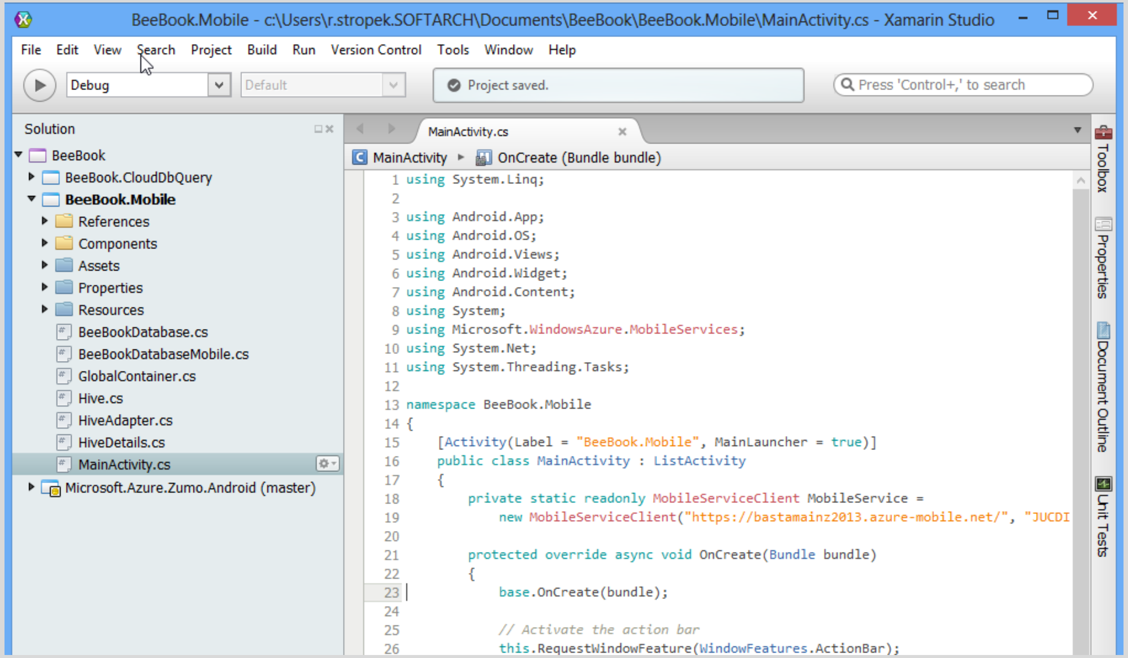
Linker

- Removes unused APIs to reduce size of app
  - See Xamarin docs for details
- Shared runtime during debugging
  - Reduces package size
  - Speedup app deployment and startup during debugging

Development

IDEs, Debugging

Xamarin Studio


BeeBook.Mobile - c:\Users\r.stropek.SOFTARCH\Documents\BeeBook\BeeBook.Mobile\MainActivity.cs - Xamarin Studio
File Edit View Search Project Build Run Version Control Tools Window Help
Debug
Default
Press 'Control+,' to search
MainActivity.cs
MainActivity ▸ OnCreate (Bundle bundle)
Toolbox
Properties
Document Outline
Unit Tests

Project saved.

Solution

- BeeBook
  - BeeBook.CloudDbQuery
  - **BeeBook.Mobile**
    - References
    - Components
    - Assets
    - Properties
    - Resources
    - BeeBookDatabase.cs
    - BeeBookDatabaseMobile.cs
    - GlobalContainer.cs
    - Hive.cs
    - HiveAdapter.cs
    - HiveDetails.cs
    - MainActivity.cs
  - Microsoft.Azure.Zumo.Android (master)

```
using System.Linq;

using Android.App;
using Android.OS;
using Android.Views;
using Android.Widget;
using Android.Content;
using System;
using Microsoft.WindowsAzure.MobileServices;
using System.Net;
using System.Threading.Tasks;

namespace BeeBook.Mobile
{
    [Activity(Label = "BeeBook.Mobile", MainLauncher = true)]
    public class MainActivity : ListActivity
    {
        private static readonly MobileServiceClient MobileService =
            new MobileServiceClient("https://bastamainz2013.azure-mobile.net/", "JUCDI

        protected override async void OnCreate(Bundle bundle)
        {
            base.OnCreate(bundle);

            // Activate the action bar
            this.RequestWindowFeature(WindowFeatures.ActionBar);
```

Works on Windows and Mac

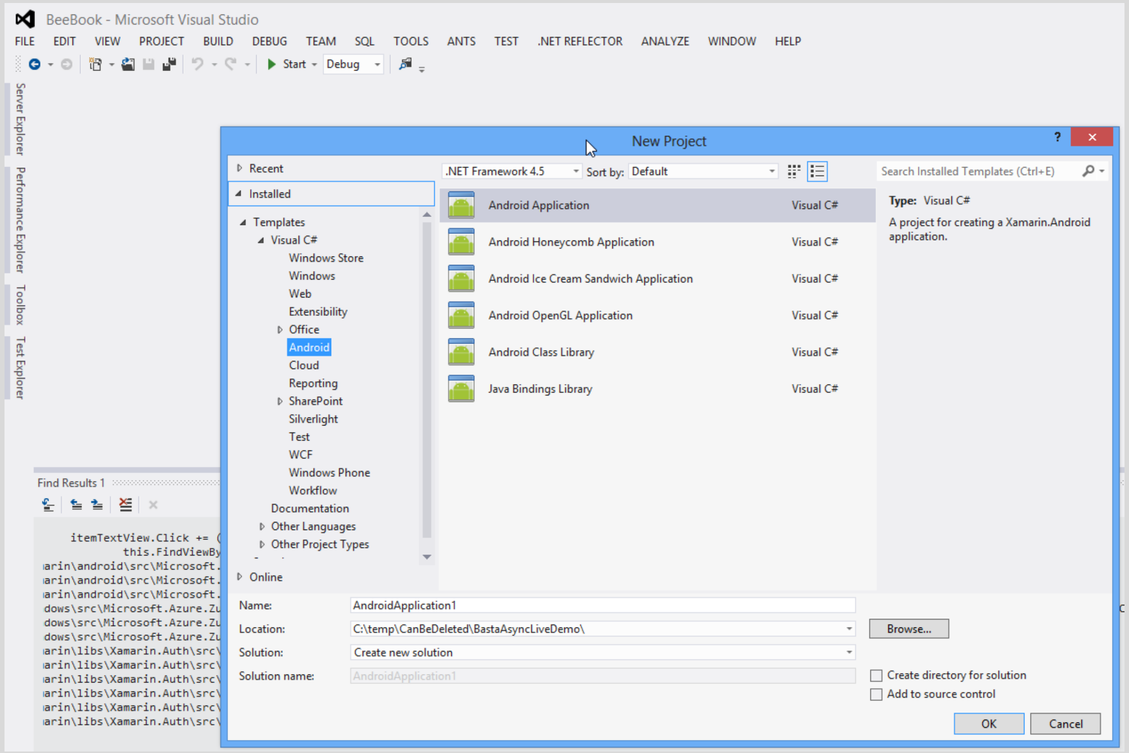
Basics similar to Visual Studio

- If you know VS, you will immediately understand it
- By far not that powerful as VS

Identical project/solution file format

- Open project/solution files in both IDEs as you need it

Visual Studio



Android Application	Visual C#
Android Honeycomb Application	Visual C#
Android Ice Cream Sandwich Application	Visual C#
Android OpenGL Application	Visual C#
Android Class Library	Visual C#
Java Bindings Library	Visual C#

Type: Visual C#

A project for creating a Xamarin.Android application.

Name: AndroidApplication1

Location: C:\temp\CanBeDeleted\BastaAsyncLiveDemo\

Solution: Create new solution

Solution name: AndroidApplication1

```
    itemTextView.Click += (
          this.FindViewBy
arin\android\src\Microsoft.
arin\android\src\Microsoft.
arin\android\src\Microsoft.
dows\src\Microsoft.Azure.Zu
dows\src\Microsoft.Azure.Zu
dows\src\Microsoft.Azure.Zu
arin\libs\Xamarin.Auth\src\
arin\libs\Xamarin.Auth\src\
arin\libs\Xamarin.Auth\src\
arin\libs\Xamarin.Auth\src\
arin\libs\Xamarin.Auth\src\
arin\libs\Xamarin.Auth\src\
```

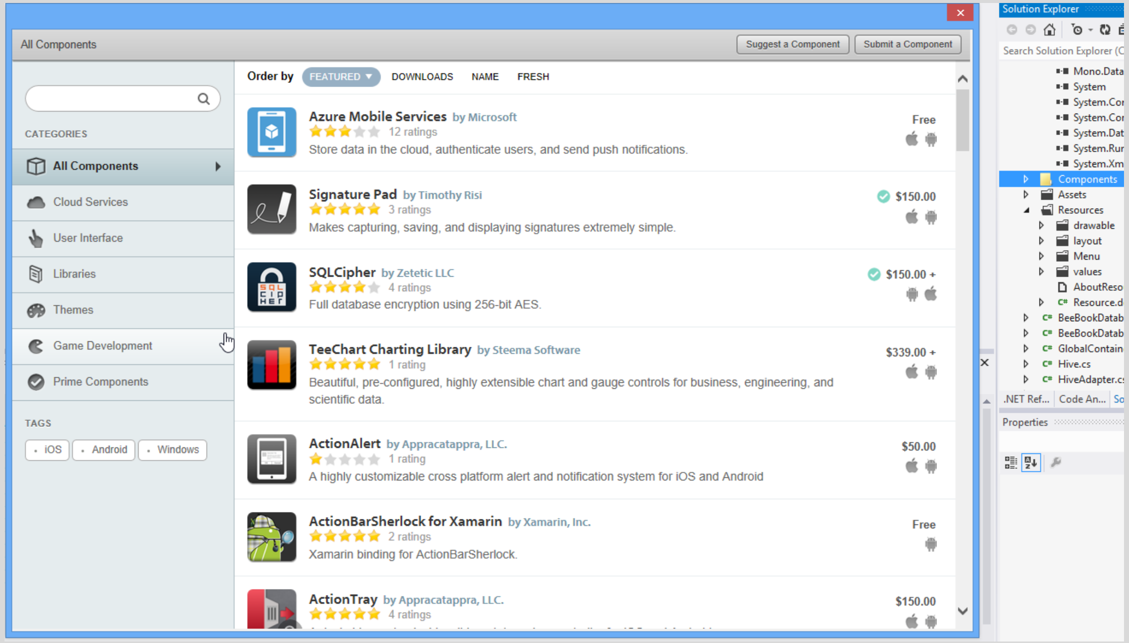
Works only on Windows

The full IDE experience you are used to
- Same C# editor
- Same UI

Full deployment and debugging support

Components

Xamarin Components



All Components

Suggest a Component | Submit a Component

CATEGORIES
- All Components
- Cloud Services
- User Interface
- Libraries
- Themes
- Game Development
- Prime Components

TAGS
iOS · Android · Windows

Order by FEATURED DOWNLOADS NAME FRESH

Azure Mobile Services by Microsoft — 12 ratings — Free
Store data in the cloud, authenticate users, and send push notifications.

Signature Pad by Timothy Risi — 3 ratings — $150.00
Makes capturing, saving, and displaying signatures extremely simple.

SQLCipher by Zetetic LLC — 4 ratings — $150.00 +
Full database encryption using 256-bit AES.

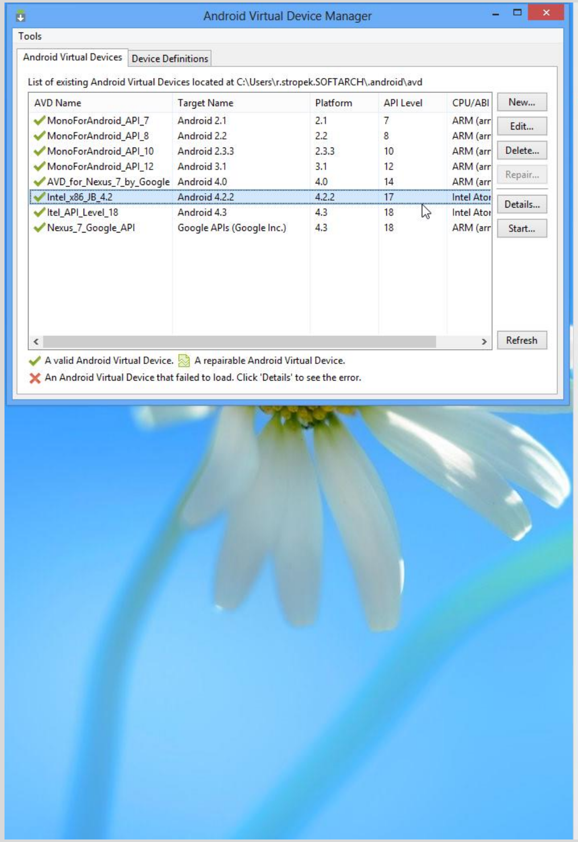
TeeChart Charting Library by Steema Software — 1 rating — $339.00 +
Beautiful, pre-configured, highly extensible chart and gauge controls for business, engineering, and scientific data.

ActionAlert by Appracatappra, LLC. — 1 rating — $50.00
A highly customizable cross platform alert and notification system for iOS and Android

ActionBarSherlock for Xamarin by Xamarin, Inc. — 2 ratings — Free
Xamarin binding for ActionBarSherlock.

ActionTray by Appracatappra, LLC. — 4 ratings — $150.00

Full list of components see [Xamarin Components website]



Android Virtual Device Manager

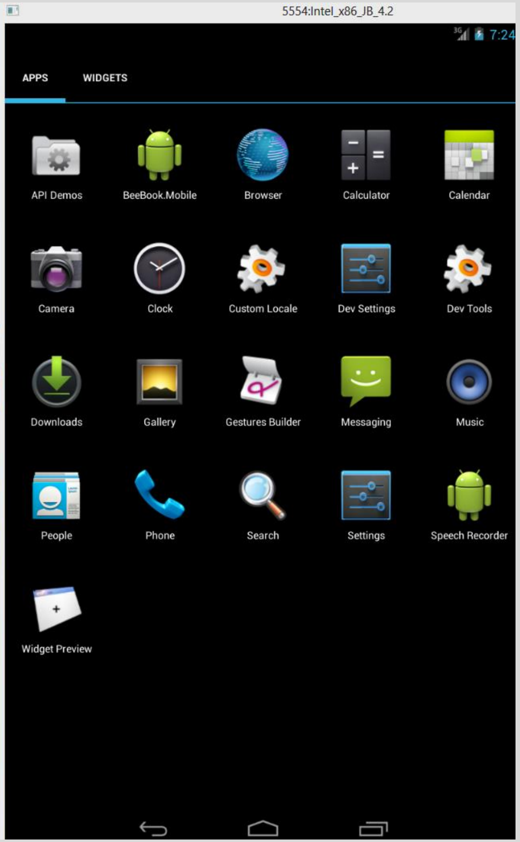
Tools

Android Virtual Devices | Device Definitions

List of existing Android Virtual Devices located at C:\Users\r.stropek.SOFTARCH\.android\avd

AVD Name	Target Name	Platform	API Level	CPU/ABI
✓ MonoForAndroid_API_7	Android 2.1	2.1	7	ARM (arr
✓ MonoForAndroid_API_8	Android 2.2	2.2	8	ARM (arr
✓ MonoForAndroid_API_10	Android 2.3.3	2.3.3	10	ARM (arr
✓ MonoForAndroid_API_12	Android 3.1	3.1	12	ARM (arr
✓ AVD_for_Nexus_7_by_Google	Android 4.0	4.0	14	ARM (arr
✓ Intel_x86_JB_4.2	Android 4.2.2	4.2.2	17	Intel Ator
✓ Itel_API_Level_18	Android 4.3	4.3	18	Intel Ator
✓ Nexus_7_Google_API	Google APIs (Google Inc.)	4.3	18	ARM (arr

✓ A valid Android Virtual Device. A repairable Android Virtual Device.
✗ An Android Virtual Device that failed to load. Click 'Details' to see the error.



Emulator

Debug and test your app

Android Device Emulator

- Not specific to Xamarin
- All features, tools, and restrictions of native Android development apply

ARM or Intel-based images

- ARM images are *very* slow
- Recommendation: Intel image with HAXM

Demo

Sample app in Xamarin Studio and Visual Studio

Hello World!

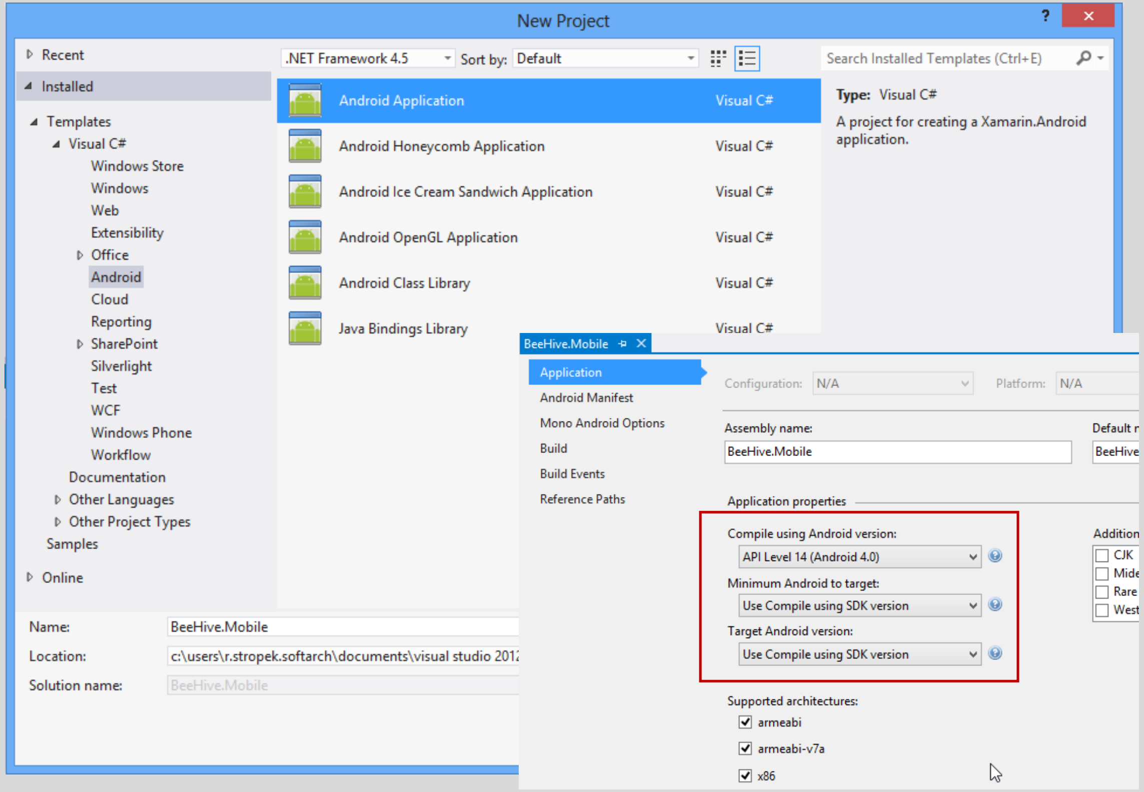
Comparing the developer environments

Debugging experience

Sample

Create Project

Create and configure project



New Project

Recent
Installed
Templates
Visual C#
Windows Store
Windows
Web
Extensibility
Office
Android
Cloud
Reporting
SharePoint
Silverlight
Test
WCF
Windows Phone
Workflow
Documentation
Other Languages
Other Project Types
Samples
Online

.NET Framework 4.5 · Sort by: Default · Search Installed Templates (Ctrl+E)

Android Application	Visual C#
Android Honeycomb Application	Visual C#
Android Ice Cream Sandwich Application	Visual C#
Android OpenGL Application	Visual C#
Android Class Library	Visual C#
Java Bindings Library	Visual C#

Type: Visual C#

A project for creating a Xamarin.Android application.

Name: BeeHive.Mobile
Location: c:\users\r.stropek.softarch\documents\visual studio 201
Solution name: BeeHive.Mobile

BeeHive.Mobile

Application
Android Manifest
Mono Android Options
Build
Build Events
Reference Paths

Configuration: N/A · Platform: N/A

Assembly name: BeeHive.Mobile
Default r: BeeHive

Application properties

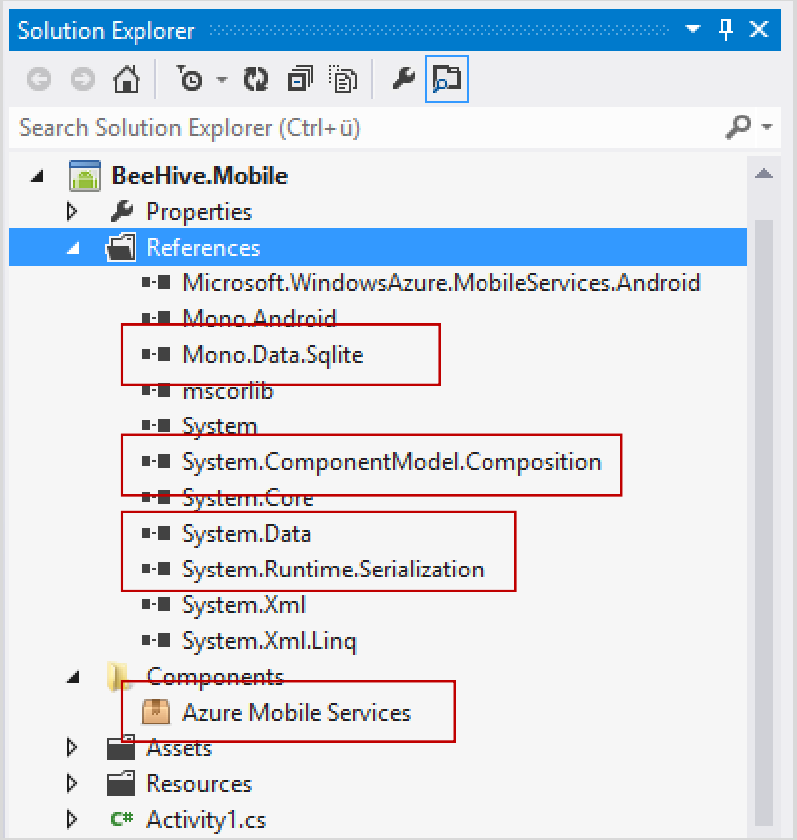
Compile using Android version: API Level 14 (Android 4.0)
Minimum Android to target: Use Compile using SDK version
Target Android version: Use Compile using SDK version

Supported architectures:
- [x] armeabi
- [x] armeabi-v7a
- [x] x86

Addition
- [ ] CJK
- [ ] Mide
- [ ] Rare
- [ ] West

Create Project

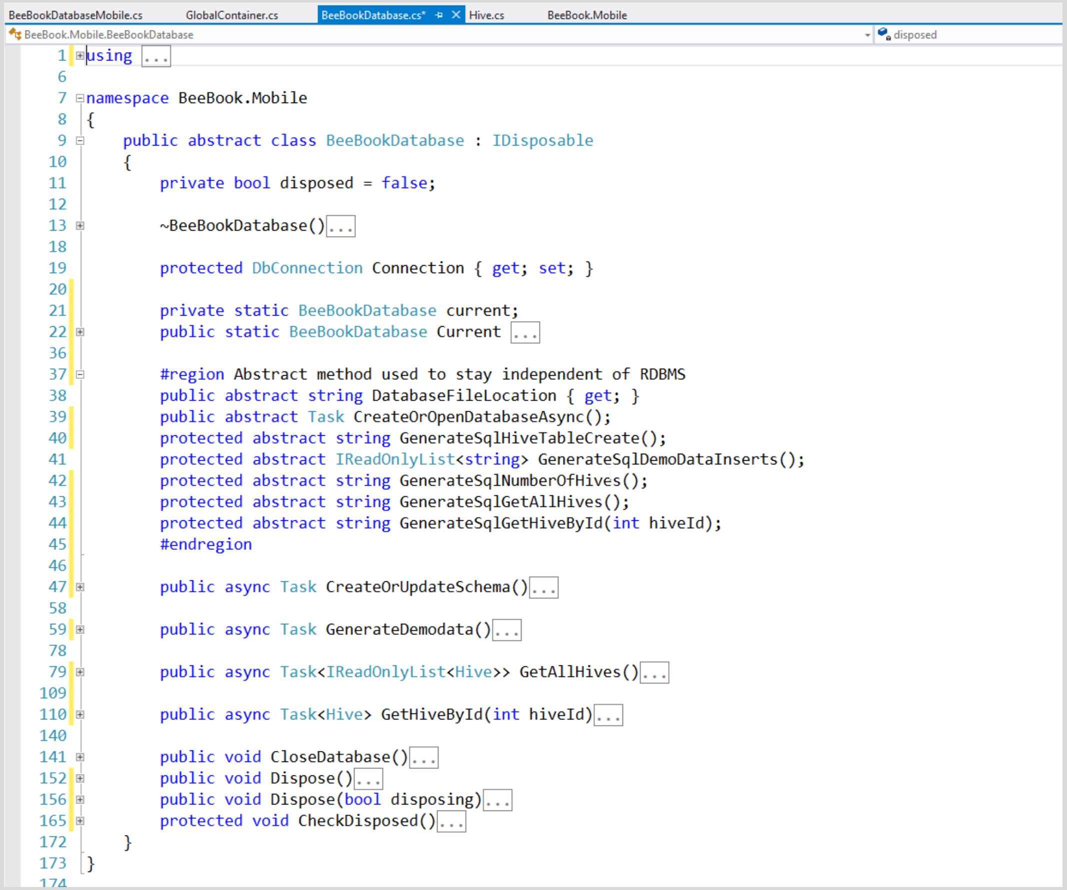
Setup project dependencies



Solution Explorer

Search Solution Explorer (Ctrl+ü)

BeeHive.Mobile
- Properties
- References
  - Microsoft.WindowsAzure.MobileServices.Android
  - Mono.Android
  - Mono.Data.Sqlite
  - mscorlib
  - System
  - System.ComponentModel.Composition
  - System.Core
  - System.Data
  - System.Runtime.Serialization
  - System.Xml
  - System.Xml.Linq
- Components
  - Azure Mobile Services
- Assets
- Resources
- Activity1.cs


BeeBookDatabaseMobile.cs GlobalContainer.cs BeeBookDatabase.cs* Hive.cs BeeBook.Mobile

BeeBook.Mobile.BeeBookDatabase disposed

```
using ...

namespace BeeBook.Mobile
{
    public abstract class BeeBookDatabase : IDisposable
    {
        private bool disposed = false;

        ~BeeBookDatabase()...

        protected DbConnection Connection { get; set; }

        private static BeeBookDatabase current;
        public static BeeBookDatabase Current ...

        #region Abstract method used to stay independent of RDBMS
        public abstract string DatabaseFileLocation { get; }
        public abstract Task CreateOrOpenDatabaseAsync();
        protected abstract string GenerateSqlHiveTableCreate();
        protected abstract IReadOnlyList<string> GenerateSqlDemoDataInserts();
        protected abstract string GenerateSqlNumberOfHives();
        protected abstract string GenerateSqlGetAllHives();
        protected abstract string GenerateSqlGetHiveById(int hiveId);
        #endregion

        public async Task CreateOrUpdateSchema()...

        public async Task GenerateDemodata()...

        public async Task<IReadOnlyList<Hive>> GetAllHives()...

        public async Task<Hive> GetHiveById(int hiveId)...

        public void CloseDatabase()...
        public void Dispose()...
        public void Dispose(bool disposing)...
        protected void CheckDisposed()...
    }
}
```

Data Access Layer

Platform-independent code

Abstract base class

- Could be in a separate class library (PCL)
- Use *DBConnection* to keep code reusable

Other strategies

- Link source files
- Use partial classes
- Use interfaces to isolate platform-specific aspects in your code
- Conditional compile
- Use patterns like MVVM to reduce amount of platform-specific code

Data Access Layer

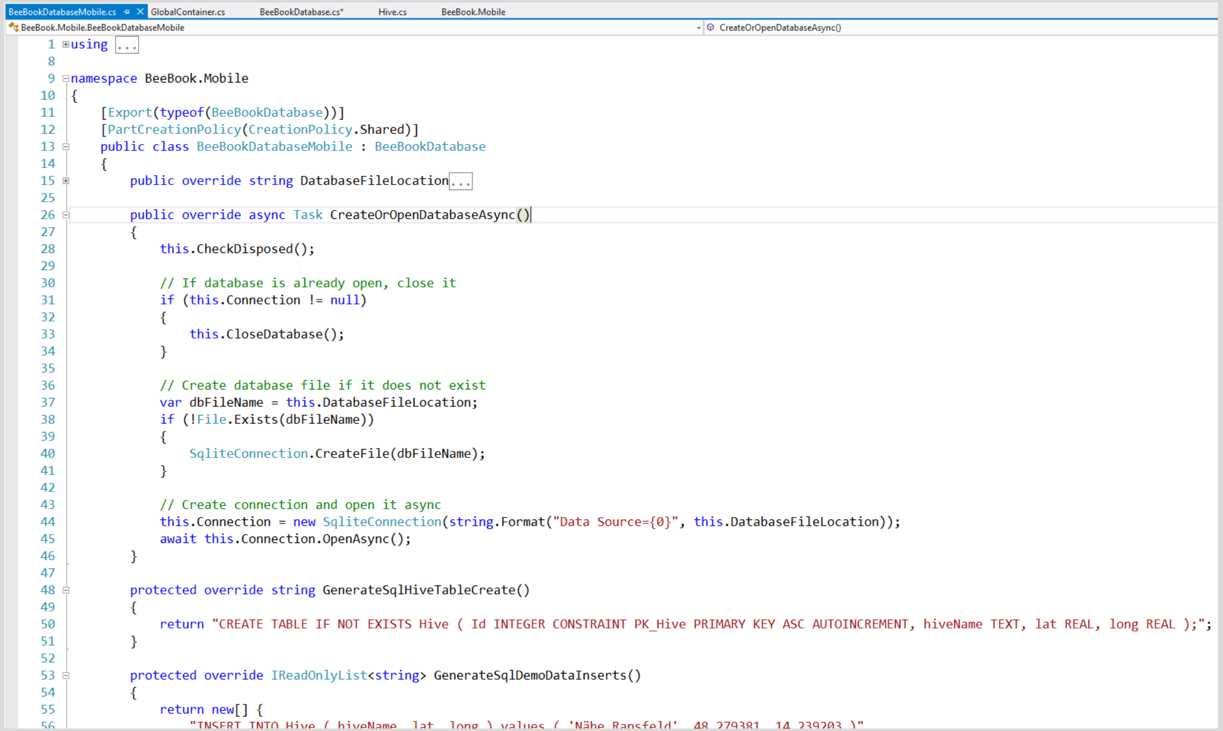
Platform-independent code

Implementation for mobile device

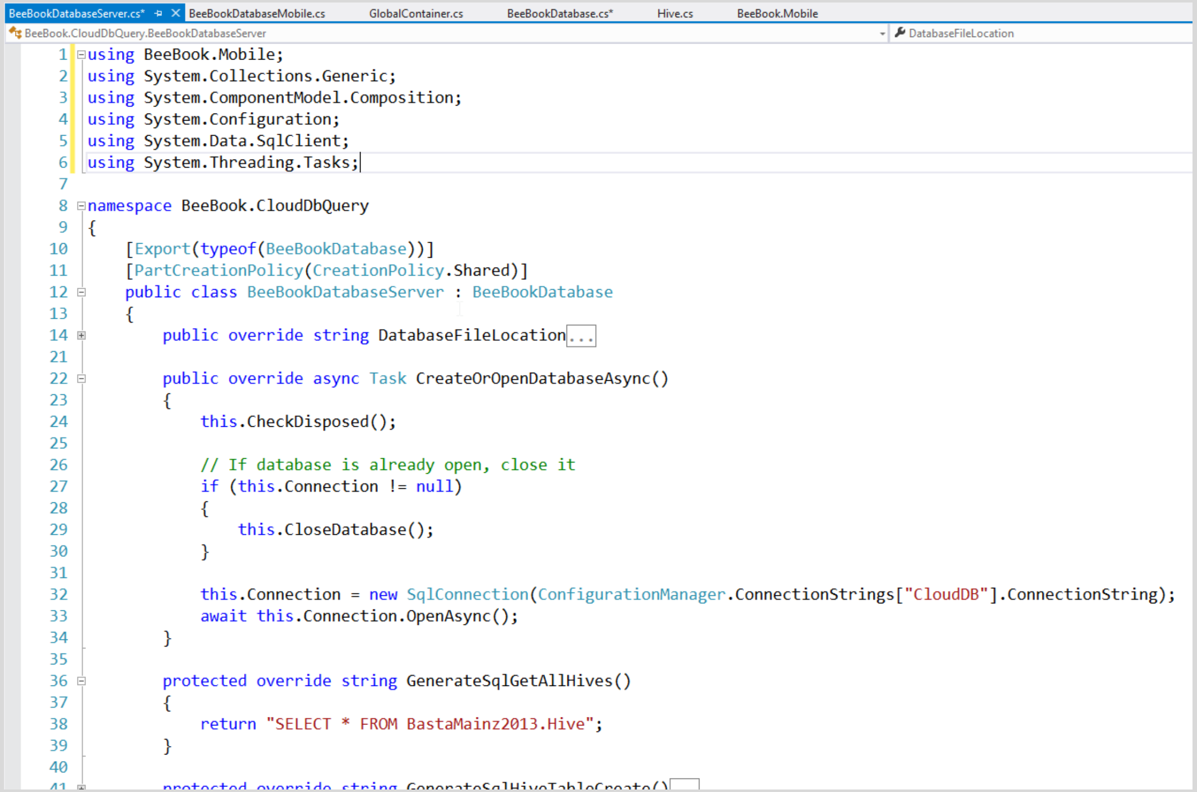
- SQLite

Note

- Full support for MEF
- Async APIs
- C# 5 *async/await*



```
using ...

namespace BeeBook.Mobile
{
    [Export(typeof(BeeBookDatabase))]
    [PartCreationPolicy(CreationPolicy.Shared)]
    public class BeeBookDatabaseMobile : BeeBookDatabase
    {
        public override string DatabaseFileLocation...

        public override async Task CreateOrOpenDatabaseAsync()
        {
            this.CheckDisposed();

            // If database is already open, close it
            if (this.Connection != null)
            {
                this.CloseDatabase();
            }

            // Create database file if it does not exist
            var dbFileName = this.DatabaseFileLocation;
            if (!File.Exists(dbFileName))
            {
                SqliteConnection.CreateFile(dbFileName);
            }

            // Create connection and open it async
            this.Connection = new SqliteConnection(string.Format("Data Source={0}", this.DatabaseFileLocation));
            await this.Connection.OpenAsync();
        }

        protected override string GenerateSqlHiveTableCreate()
        {
            return "CREATE TABLE IF NOT EXISTS Hive ( Id INTEGER CONSTRAINT PK_Hive PRIMARY KEY ASC AUTOINCREMENT, hiveName TEXT, lat REAL, long REAL );";
        }

        protected override IReadOnlyList<string> GenerateSqlDemoDataInserts()
        {
            return new[] {
                "INSERT INTO Hive ( hiveName, lat, long ) values ( 'Nähe Ransfeld', 48.279381, 14.239203 )"
```



```csharp
using BeeBook.Mobile;
using System.Collections.Generic;
using System.ComponentModel.Composition;
using System.Configuration;
using System.Data.SqlClient;
using System.Threading.Tasks;

namespace BeeBook.CloudDbQuery
{
    [Export(typeof(BeeBookDatabase))]
    [PartCreationPolicy(CreationPolicy.Shared)]
    public class BeeBookDatabaseServer : BeeBookDatabase
    {
        public override string DatabaseFileLocation...

        public override async Task CreateOrOpenDatabaseAsync()
        {
            this.CheckDisposed();

            // If database is already open, close it
            if (this.Connection != null)
            {
                this.CloseDatabase();
            }

            this.Connection = new SqlConnection(ConfigurationManager.ConnectionStrings["CloudDB"].ConnectionString);
            await this.Connection.OpenAsync();
        }

        protected override string GenerateSqlGetAllHives()
        {
            return "SELECT * FROM BastaMainz2013.Hive";
        }

        protected override string GenerateSqlHiveTableCreate()
```

Data Access Layer

Platform-independent code

Implementation for

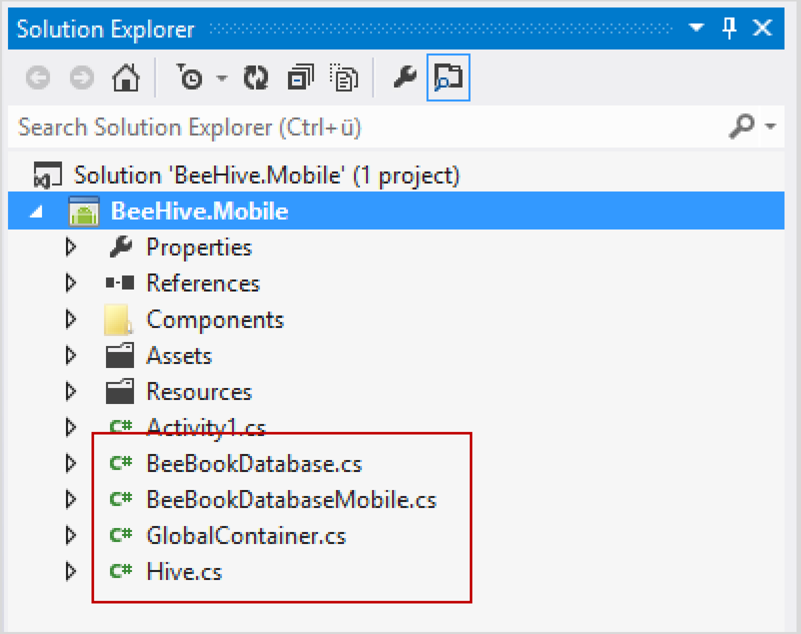
SQL Server

Windows Azure SQL Database

Add Data Access

Add existing items

Mobile DAL
 implementation



Solution Explorer

Search Solution Explorer (Ctrl+ü)

Solution 'BeeHive.Mobile' (1 project)

BeeHive.Mobile

Properties

References

Components

Assets

Resources

Activity1.cs

BeeBookDatabase.cs

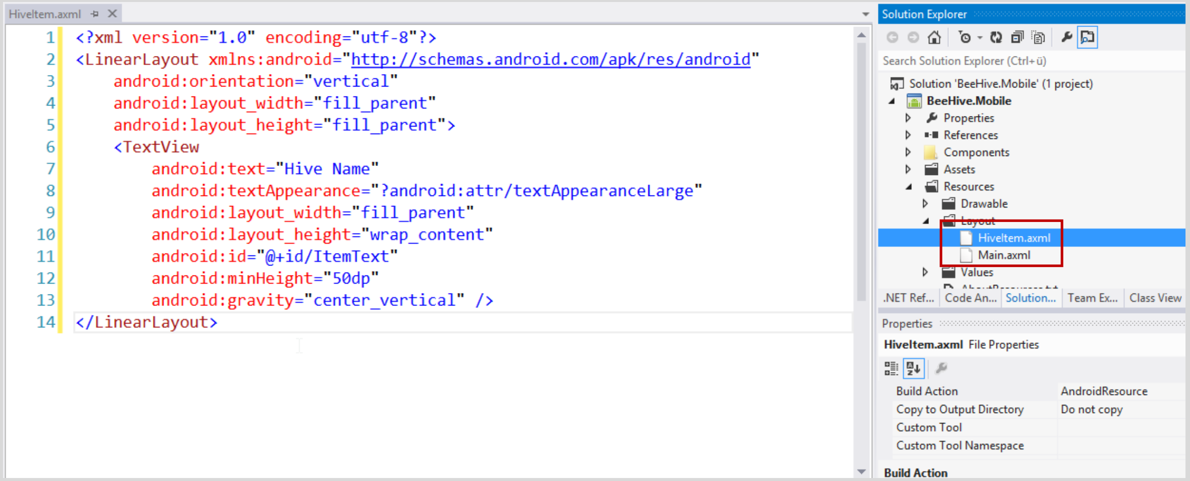
BeeBookDatabaseMobile.cs

GlobalContainer.cs

Hive.cs

Add UI

UI for each list item



HiveItem.axml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:text="Hive Name"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/ItemText"
        android:minHeight="50dp"
        android:gravity="center_vertical" />
</LinearLayout>
```

Solution Explorer

Search Solution Explorer (Ctrl+ü)

- Solution 'BeeHive.Mobile' (1 project)
  - BeeHive.Mobile
    - Properties
    - References
    - Components
    - Assets
    - Resources
      - Drawable
      - Layout
        - HiveItem.axml
        - Main.axml
      - Values

.NET Ref... | Code An... | Solution... | Team Ex... | Class View

Properties

HiveItem.axml File Properties

Build Action	AndroidResource
Copy to Output Directory	Do not copy
Custom Tool	
Custom Tool Namespace	

Build Action

BASTA!

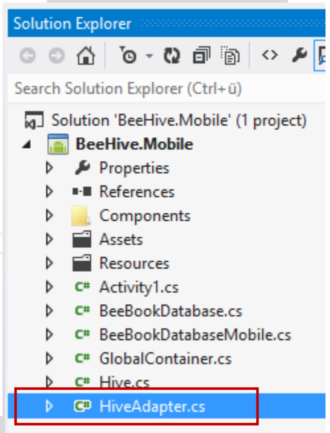
Add Adapter

Data Source for *ListView*

Note

- Use of *async/await*
- Use of *Intent*

```
public class HiveAdapter : BaseAdapter<Hive>
{
    private IReadOnlyList<Hive> items = new List<Hive>();
    private readonly LayoutInflater inflater;

    public HiveAdapter(Context context)
    {
        this.inflater = (LayoutInflater)context.GetSystemService(Context.LayoutInflaterService);
        this.RefreshAsync();
    }

    public override bool HasStableIds { get { return true; } }
    public override int Count { get { return this.items.Count; } }
    public override Hive this[int position] { get { return this.items[position]; } }
    public override long GetItemId(int position) { return this.items[position].Id; }

    public override View GetView(int position, View convertView, ViewGroup parent)
    {
        var item = this.items[position];

        var view = this.inflater.Inflate(Resource.Layout.HiveItem, null);
        var itemTextView = view.FindViewById<TextView>(Resource.Id.ItemText);
        itemTextView.Text = item.HiveName;

        //itemTextView.Click += (o, e) =>
        //{
        //  var hiveDetailsActivity = new Intent(this.inflater.Context, typeof(HiveDetails));
        //  hiveDetailsActivity.PutExtra("Id", item.Id);
        //  this.inflater.Context.StartActivity(hiveDetailsActivity);
        //};

        return view;
    }

    public async void RefreshAsync()
    {
        this.items = await BeeBookDatabase.Current.GetAllHives();
        this.NotifyDataSetChanged();
    }
}
```



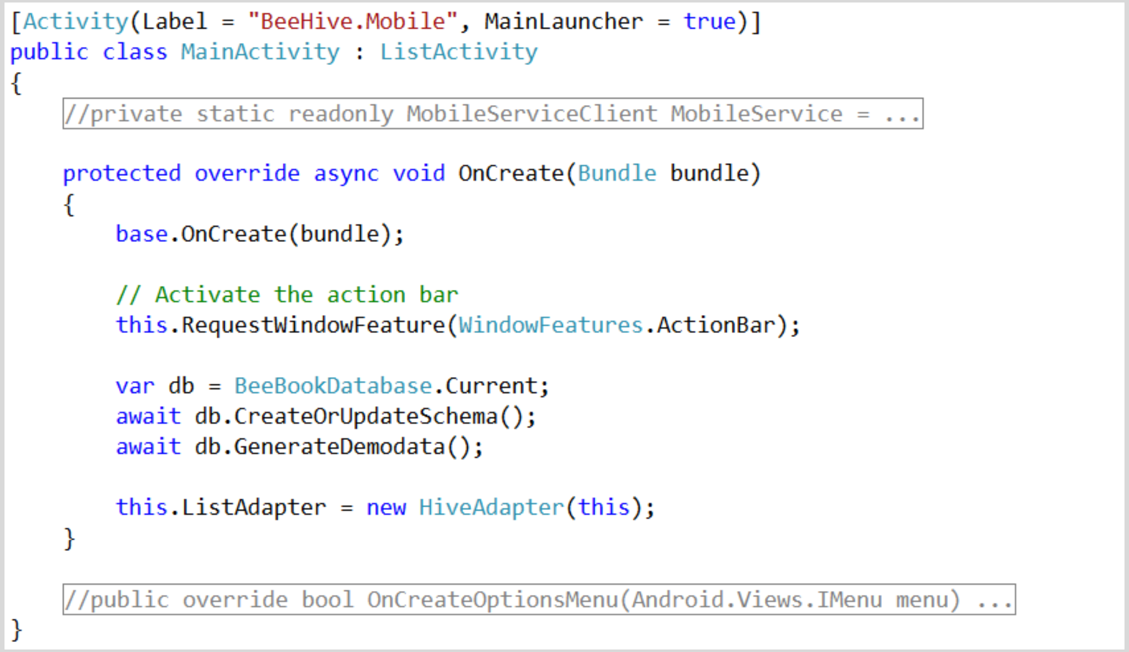
Solution Explorer

Search Solution Explorer (Ctrl+ü)

- Solution 'BeeHive.Mobile' (1 project)
  - **BeeHive.Mobile**
    - Properties
    - References
    - Components
    - Assets
    - Resources
    - Activity1.cs
    - BeeBookDatabase.cs
    - BeeBookDatabaseMobile.cs
    - GlobalContainer.cs
    - Hive.cs
    - HiveAdapter.cs

Main Activity

Entry point for your app



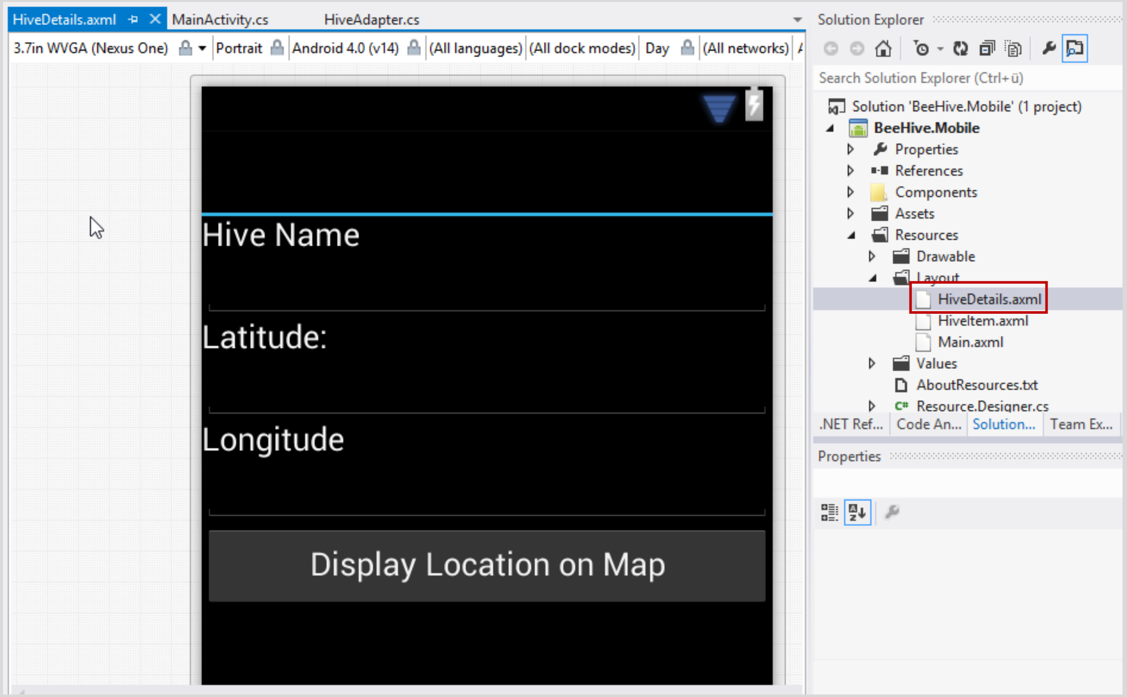
```
[Activity(Label = "BeeHive.Mobile", MainLauncher = true)]
public class MainActivity : ListActivity
{
    //private static readonly MobileServiceClient MobileService = ...

    protected override async void OnCreate(Bundle bundle)
    {
        base.OnCreate(bundle);

        // Activate the action bar
        this.RequestWindowFeature(WindowFeatures.ActionBar);

        var db = BeeBookDatabase.Current;
        await db.CreateOrUpdateSchema();
        await db.GenerateDemodata();

        this.ListAdapter = new HiveAdapter(this);
    }

    //public override bool OnCreateOptionsMenu(Android.Views.IMenu menu) ...
}
```

Note

- Renamed *Activity1* to *MainActivity*
- Change base class to *ListActivity*

UI for Details

Add second Activity



HiveDetails.axml | MainActivity.cs | HiveAdapter.cs

3.7in WVGA (Nexus One) | Portrait | Android 4.0 (v14) | (All languages) | (All dock modes) | Day | (All networks)

Hive Name

Latitude:

Longitude

Display Location on Map

Solution Explorer

Search Solution Explorer (Ctrl+ü)

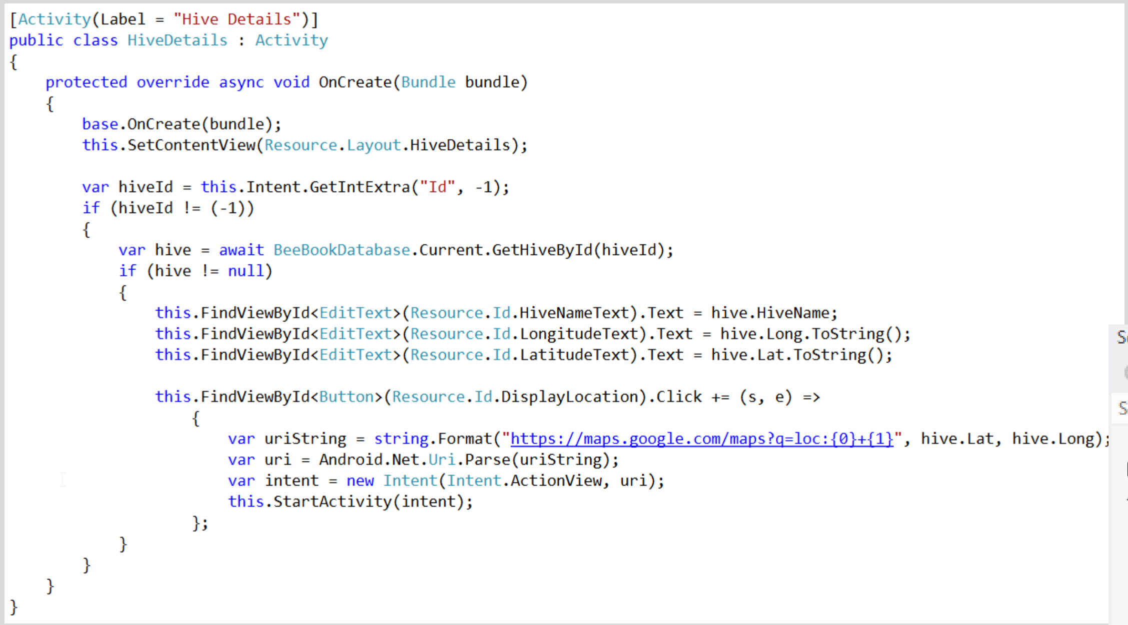
```
Solution 'BeeHive.Mobile' (1 project)
  BeeHive.Mobile
    Properties
    References
    Components
    Assets
    Resources
      Drawable
      Layout
        HiveDetails.axml
        HiveItem.axml
        Main.axml
      Values
      AboutResources.txt
      Resource.Designer.cs
```

UI for Details

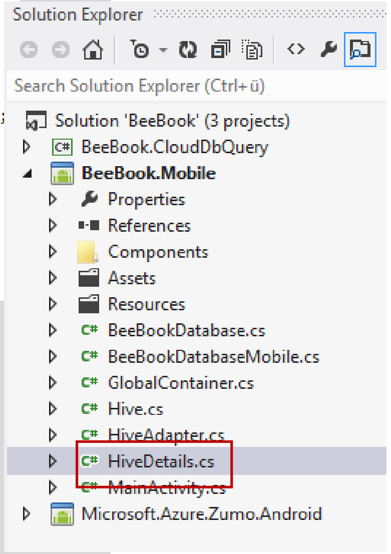
Add second Activity

Note

Intent to launch external program (browser)



```
[Activity(Label = "Hive Details")]
public class HiveDetails : Activity
{
    protected override async void OnCreate(Bundle bundle)
    {
        base.OnCreate(bundle);
        this.SetContentView(Resource.Layout.HiveDetails);

        var hiveId = this.Intent.GetIntExtra("Id", -1);
        if (hiveId != (-1))
        {
            var hive = await BeeBookDatabase.Current.GetHiveById(hiveId);
            if (hive != null)
            {
                this.FindViewById<EditText>(Resource.Id.HiveNameText).Text = hive.HiveName;
                this.FindViewById<EditText>(Resource.Id.LongitudeText).Text = hive.Long.ToString();
                this.FindViewById<EditText>(Resource.Id.LatitudeText).Text = hive.Lat.ToString();

                this.FindViewById<Button>(Resource.Id.DisplayLocation).Click += (s, e) =>
                    {
                        var uriString = string.Format("https://maps.google.com/maps?q=loc:{0}+{1}", hive.Lat, hive.Long);
                        var uri = Android.Net.Uri.Parse(uriString);
                        var intent = new Intent(Intent.ActionView, uri);
                        this.StartActivity(intent);
                    };
            }
        }
    }
}
```



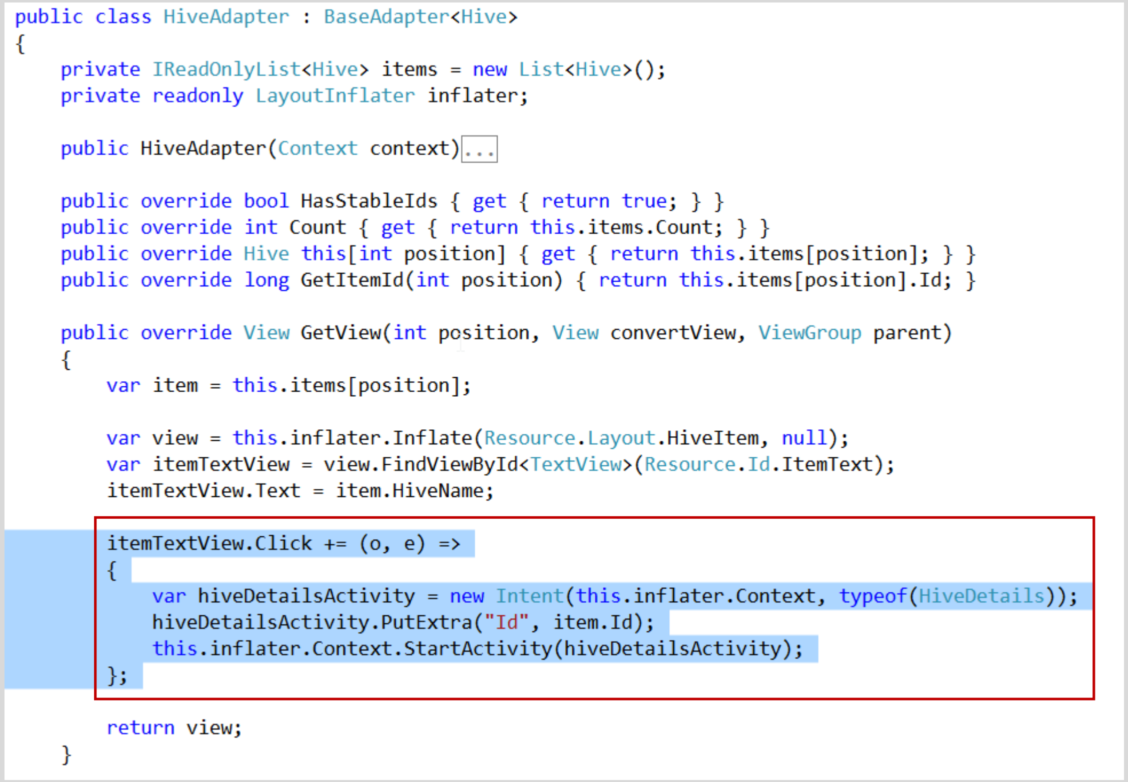
Solution Explorer

Search Solution Explorer (Ctrl+ü)

- Solution 'BeeBook' (3 projects)
  - BeeBook.CloudDbQuery
  - **BeeBook.Mobile**
    - Properties
    - References
    - Components
    - Assets
    - Resources
    - BeeBookDatabase.cs
    - BeeBookDatabaseMobile.cs
    - GlobalContainer.cs
    - Hive.cs
    - HiveAdapter.cs
    - HiveDetails.cs
    - MainActivity.cs
  - Microsoft.Azure.Zumo.Android

UI for Details

Add second Activity



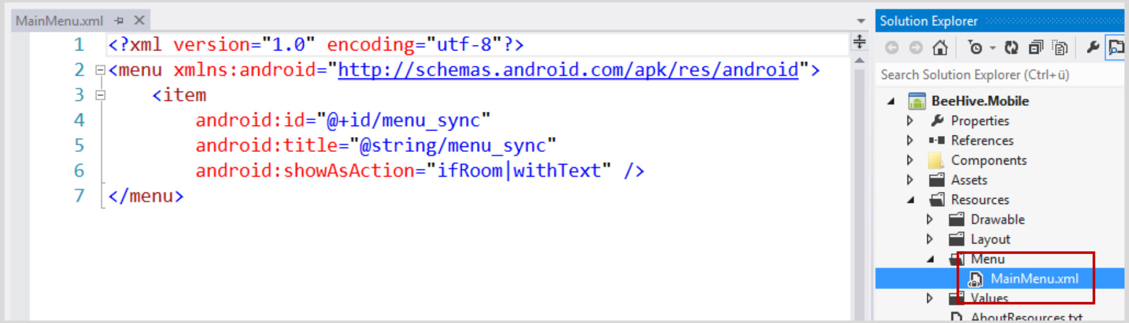
```
public class HiveAdapter : BaseAdapter<Hive>
{
    private IReadOnlyList<Hive> items = new List<Hive>();
    private readonly LayoutInflater inflater;

    public HiveAdapter(Context context)...

    public override bool HasStableIds { get { return true; } }
    public override int Count { get { return this.items.Count; } }
    public override Hive this[int position] { get { return this.items[position]; } }
    public override long GetItemId(int position) { return this.items[position].Id; }

    public override View GetView(int position, View convertView, ViewGroup parent)
    {
        var item = this.items[position];

        var view = this.inflater.Inflate(Resource.Layout.HiveItem, null);
        var itemTextView = view.FindViewById<TextView>(Resource.Id.ItemText);
        itemTextView.Text = item.HiveName;

        itemTextView.Click += (o, e) =>
        {
            var hiveDetailsActivity = new Intent(this.inflater.Context, typeof(HiveDetails));
            hiveDetailsActivity.PutExtra("Id", item.Id);
            this.inflater.Context.StartActivity(hiveDetailsActivity);
        };

        return view;
    }
```

Note

- Intent to launch external program (browser)

Action Bar

Add menu for action bar



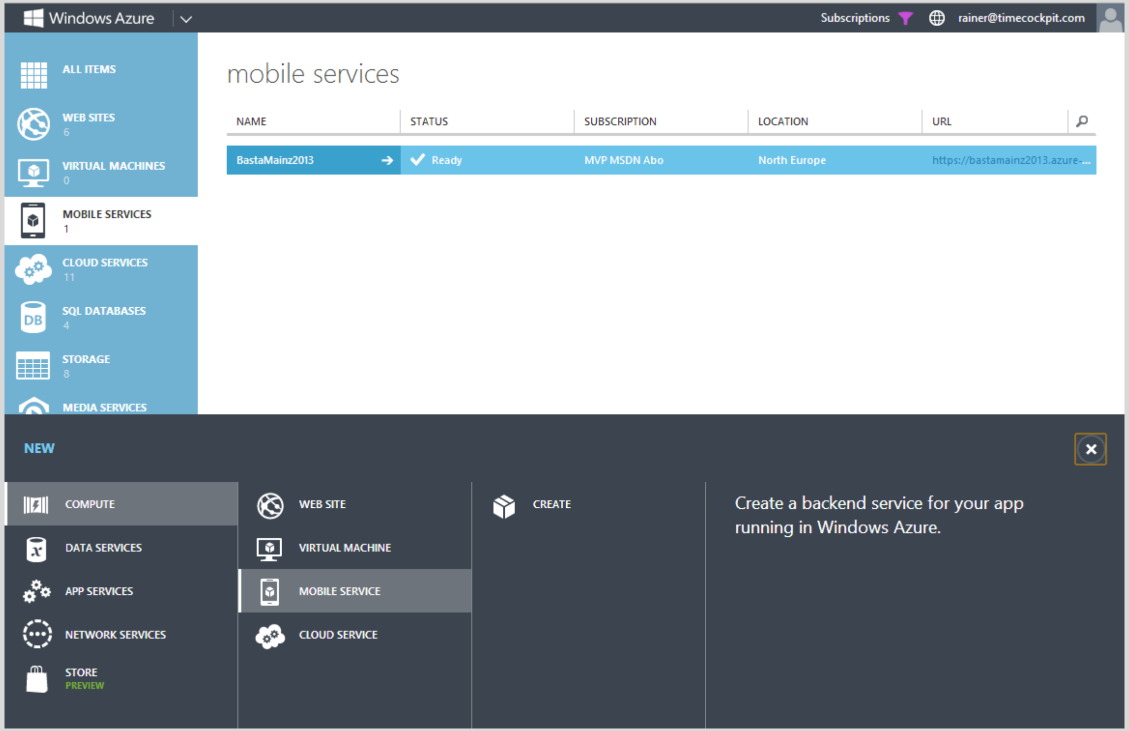
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/menu_sync"
        android:title="@string/menu_sync"
        android:showAsAction="ifRoom|withText" />
</menu>
```

Azure Mobile Service

Backend for app in the cloud

Create Azure Mobile Service

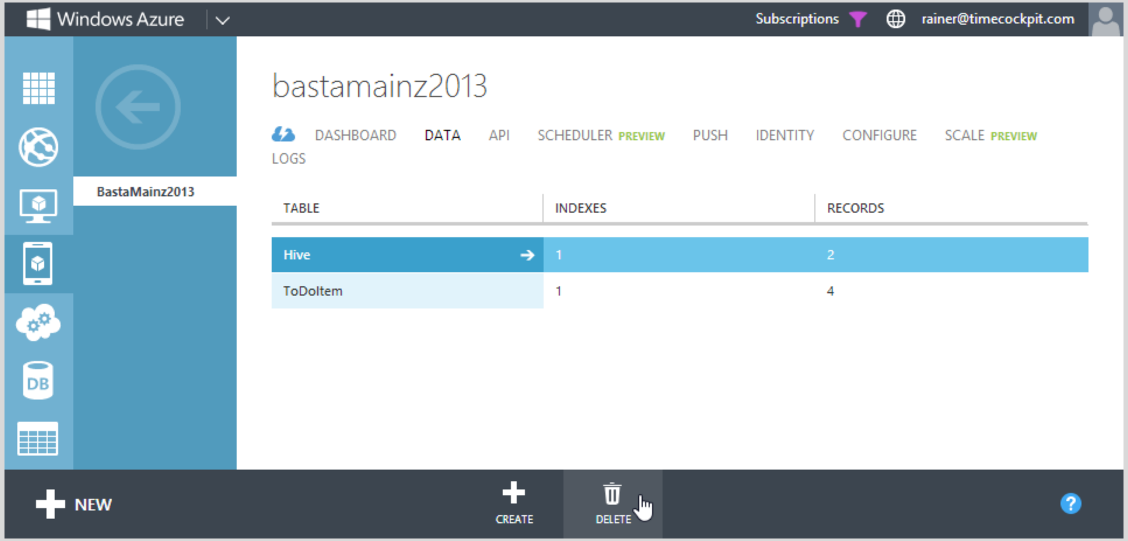
- In the background your data is stored in SQL Server



mobile services

NAME	STATUS	SUBSCRIPTION	LOCATION	URL
BastaMainz2013	✓ Ready	MVP MSDN Abo	North Europe	https://bastamainz2013.azure-...

Create a backend service for your app running in Windows Azure.


Windows Azure | Subscriptions | rainer@timecockpit.com

bastamainz2013

DASHBOARD DATA API SCHEDULER PREVIEW PUSH IDENTITY CONFIGURE SCALE PREVIEW LOGS

BastaMainz2013

TABLE	INDEXES	RECORDS
Hive	1	2
ToDoItem	1	4

Azure Mobile Service

Backend for app in the cloud

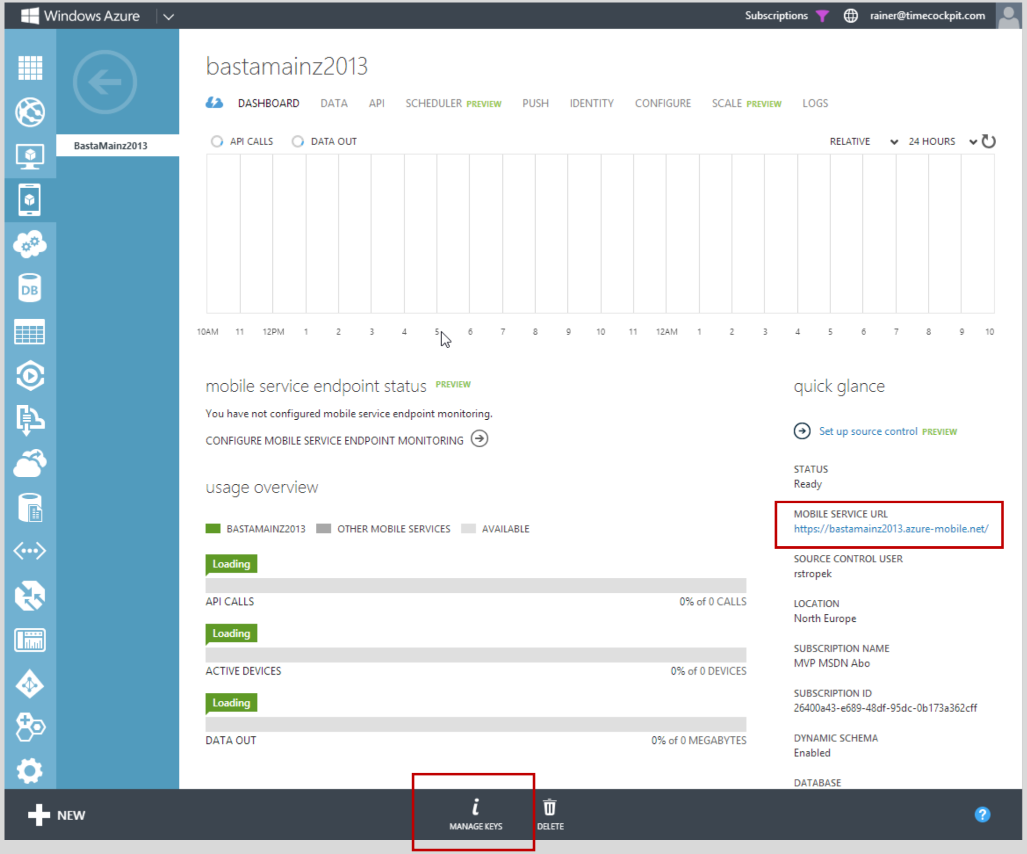
Add table *Hive*

- In the background you are creating a table in SQL Server
- Mobile Services does not need a schema → no need to create columns in the table

Azure Mobile Service

Backend for app in the cloud

Get URL and application key



bastamainz2013

mobile service endpoint status PREVIEW

You have not configured mobile service endpoint monitoring.

CONFIGURE MOBILE SERVICE ENDPOINT MONITORING

usage overview

🟩 BASTAMAINZ2013 ▨ OTHER MOBILE SERVICES ▢ AVAILABLE

API CALLS	0% of 0 CALLS
ACTIVE DEVICES	0% of 0 DEVICES
DATA OUT	0% of 0 MEGABYTES

quick glance

Set up source control PREVIEW

STATUS
Ready

MOBILE SERVICE URL
https://bastamainz2013.azure-mobile.net/

SOURCE CONTROL USER
rstropek

LOCATION
North Europe

SUBSCRIPTION NAME
MVP MSDN Abo

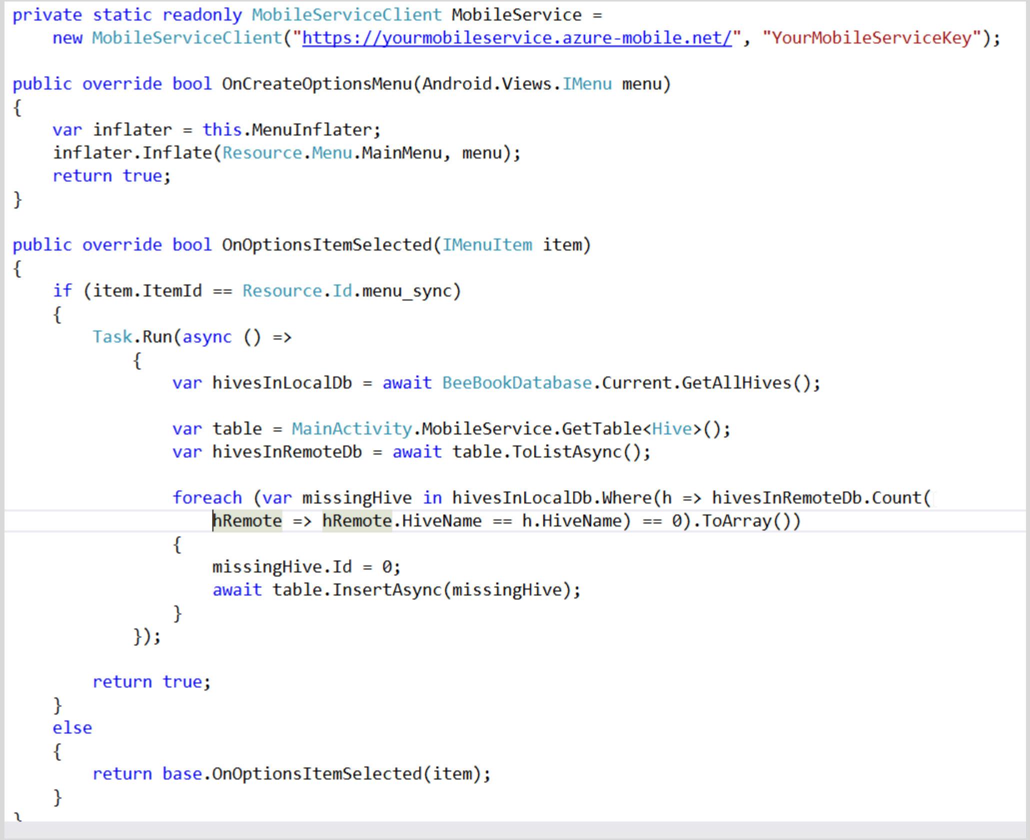
SUBSCRIPTION ID
26400a43-e689-48df-95dc-0b173a362cff

DYNAMIC SCHEMA
Enabled

DATABASE

Add Sync Code

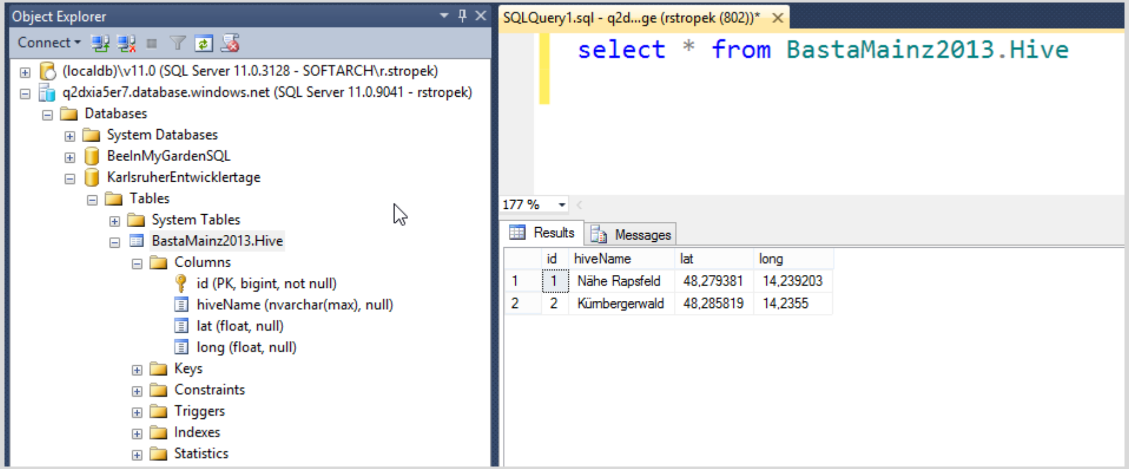
Sync triggered by action bar button



```
private static readonly MobileServiceClient MobileService =
    new MobileServiceClient("https://yourmobileservice.azure-mobile.net/", "YourMobileServiceKey");

public override bool OnCreateOptionsMenu(Android.Views.IMenu menu)
{
    var inflater = this.MenuInflater;
    inflater.Inflate(Resource.Menu.MainMenu, menu);
    return true;
}

public override bool OnOptionsItemSelected(IMenuItem item)
{
    if (item.ItemId == Resource.Id.menu_sync)
    {
        Task.Run(async () =>
            {
                var hivesInLocalDb = await BeeBookDatabase.Current.GetAllHives();

                var table = MainActivity.MobileService.GetTable<Hive>();
                var hivesInRemoteDb = await table.ToListAsync();

                foreach (var missingHive in hivesInLocalDb.Where(h => hivesInRemoteDb.Count(
                    hRemote => hRemote.HiveName == h.HiveName) == 0).ToArray())
                {
                    missingHive.Id = 0;
                    await table.InsertAsync(missingHive);
                }
            });

        return true;
    }
    else
    {
        return base.OnOptionsItemSelected(item);
    }
```

Check Result

Run the app and check result of sync in *Windows Azure SQL Database*



```
select * from BastaMainz2013.Hive
```

	id	hiveName	lat	long
1	1	Nähe Rapsfeld	48,279381	14,239203
2	2	Kürnbergerwald	48,285819	14,2355

Summary

Summary

- Great to bring existing C# knowledge to mobile platforms
  Existing business logic C# code might be reused
  Write once run anywhere is true for most business logic code
- You still have to learn and understand the platform
  Activities, Intents, Services, Adapters, Android SDK, etc.
- No or little code sharing for UI markup/code
  Can be maximized using MVVM approach

BASTA 2013 – C# Workshop

F&A

Danke für euer Kommen

Rainer Stropek

software architects gmbh

Mail	rainer@timecockpit.com
Web	http://www.timecockpit.com
Twitter	@rstropek

time cockpit
Saves the day.