Cloud-Based Real-Time Round-Trip 2D Revit Model Editing on any Mobile Device

Jeremy Tammik
Principal Developer Consultant

Class summary

- Architect a cloud-based data repository using NoSQL and Apache CouchDB
- Implement server-side scripting to display and edit 2D graphical data in the browser on a mobile device
- Understand the JavaScript implementation using jquery, db and Raphaël to generate and drive the HTML and SVG room editor
- Use the Revit API to determine room and family instance 2D boundary polygons and the Idling event for real-time BIM updates

Key learning objectives

At the end of this class, you will have

- Understood the architecture of a cloud-based data repository using NoSQL and Apache CouchDB
- Seen the server-side scripting to display and edit 2D graphical data in the browser on a mobile device
- Understood the JavaScript implementation using jquery, db and Raphaël to generate and drive the HTML and SVG room editor
- Seen the use of the Revit API to determine room and family instance 2D boundary polygons and the Idling event for real-time BIM updates

Quotes on Three Fundamental Aspects

- Lazy
  - ... develop the three great virtues of a programmer: laziness, impatience, and hubris – *Larry Wall*
- Simple
  - Simplicity is the ultimate sophistication – *Leonardo da Vinci*
  - There is no greatness where there is no simplicity – *Leo Tolstoy*
  - KISS
- Perfect
  - Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away – *Antoine de Saint-Exupéry*

About the Presenter

Jeremy Tammik
Principal Developer Consultant
Developer Technical Services
EMEA, Autodesk SARL

Jeremy is a member of the AEC workgroup of the Autodesk Developer Network ADN team, providing developer support, training, conference presentations, and blogging on the Revit API.

He joined Autodesk in 1988 as the technology evangelist responsible for European developer support to lecture, consult, and support AutoCAD application developers in Europe, the U.S., Australia, and Africa. He was a co-founder of ADGE, the AutoCAD Developer Group Europe, and a prolific author on AutoCAD application development. He left Autodesk in 1994 to work as an HVAC application developer, and then rejoined the company in 2005.

Jeremy graduated in mathematics and physics in Germany, worked as a teacher and translator, then as a C ++ programmer on early GUI and multitasking projects. He is fluent in six European languages, vegetarian, has four kids, plays the flute, likes reading, travelling, theatre improvisation, yoga, carpentry, loves mountains, oceans, sports, dancing, and especially climbing.

Data Source, Repository and Consumer Client

- BIM – Building Information Model
- Cloud-based data repository
- 2D rendering on mobile device

Real-time Editing Triggers Database and BIM Update

- Graphical room editor on mobile device
- Update cloud database
- Reflect real-time changes in BIM

Base Technologies

- BIM – Revit
- Data repository – NoSQL
- Rendering and editing – HTML and SVG

Implementation Environment

- BIM – Revit .NET add-in
- Database – Apache CouchDB
- JavaScript – jquery, db, Raphaël, SVG, HTML

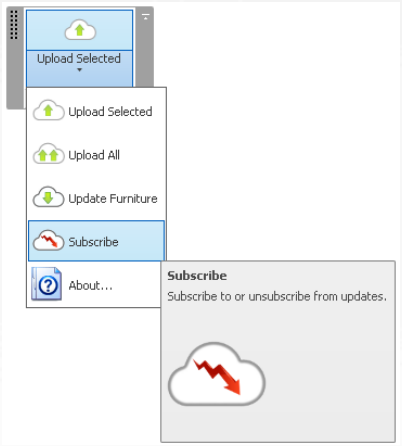
Simpler Still

- Same origin policy
- Server-side scripting
- Two components instead of three

The Two and Only Projects

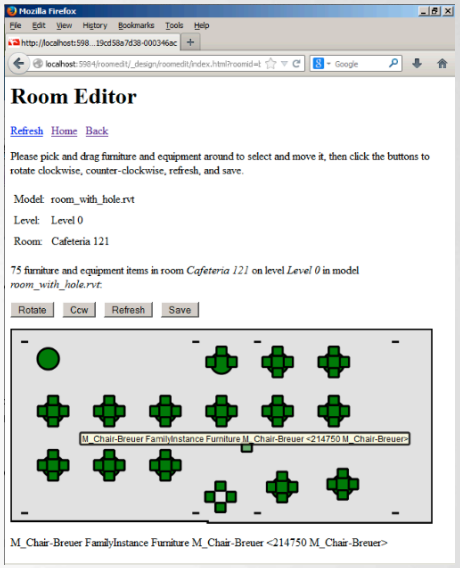
- Revit add-in
- CouchDB database

Revit Add-in



- Determine 2D boundary polygon loops
- Upload model to database
- Download changes from database
- Subscribe to real-time updates using Idling event
  - Let's look at the exciting cloud stuff first
  - Return to the add-in afterwards

Live Demonstration

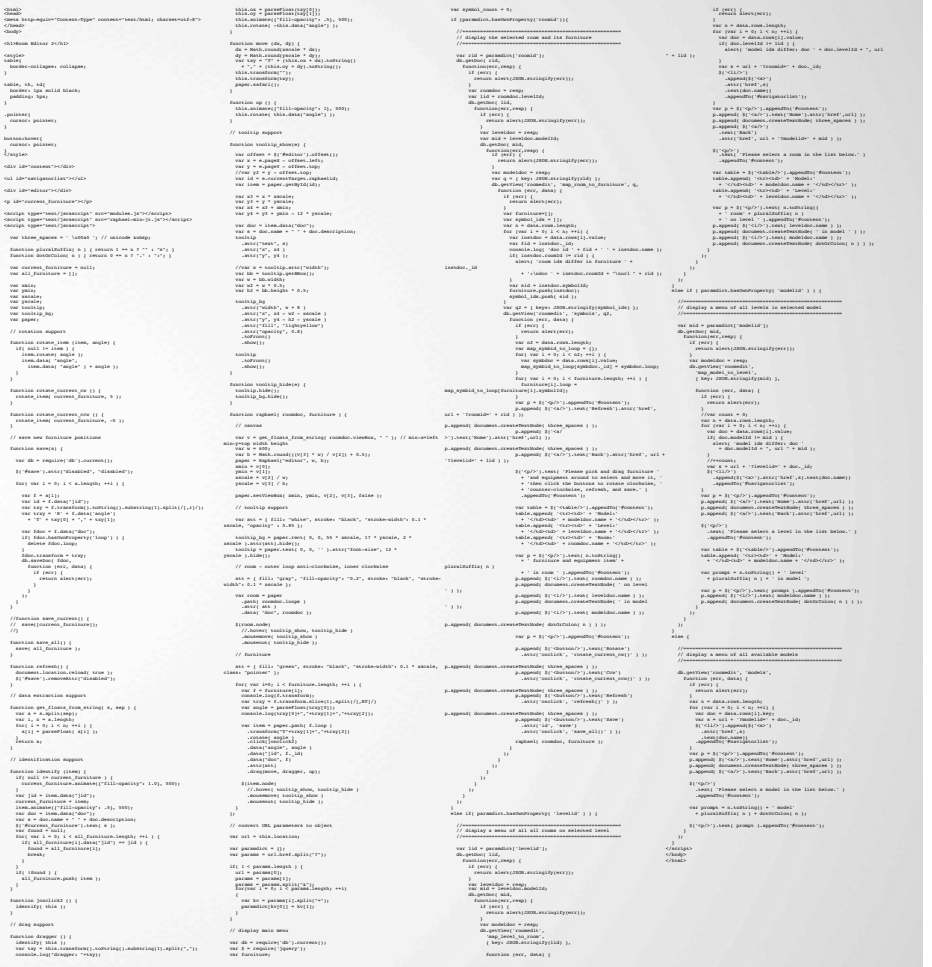


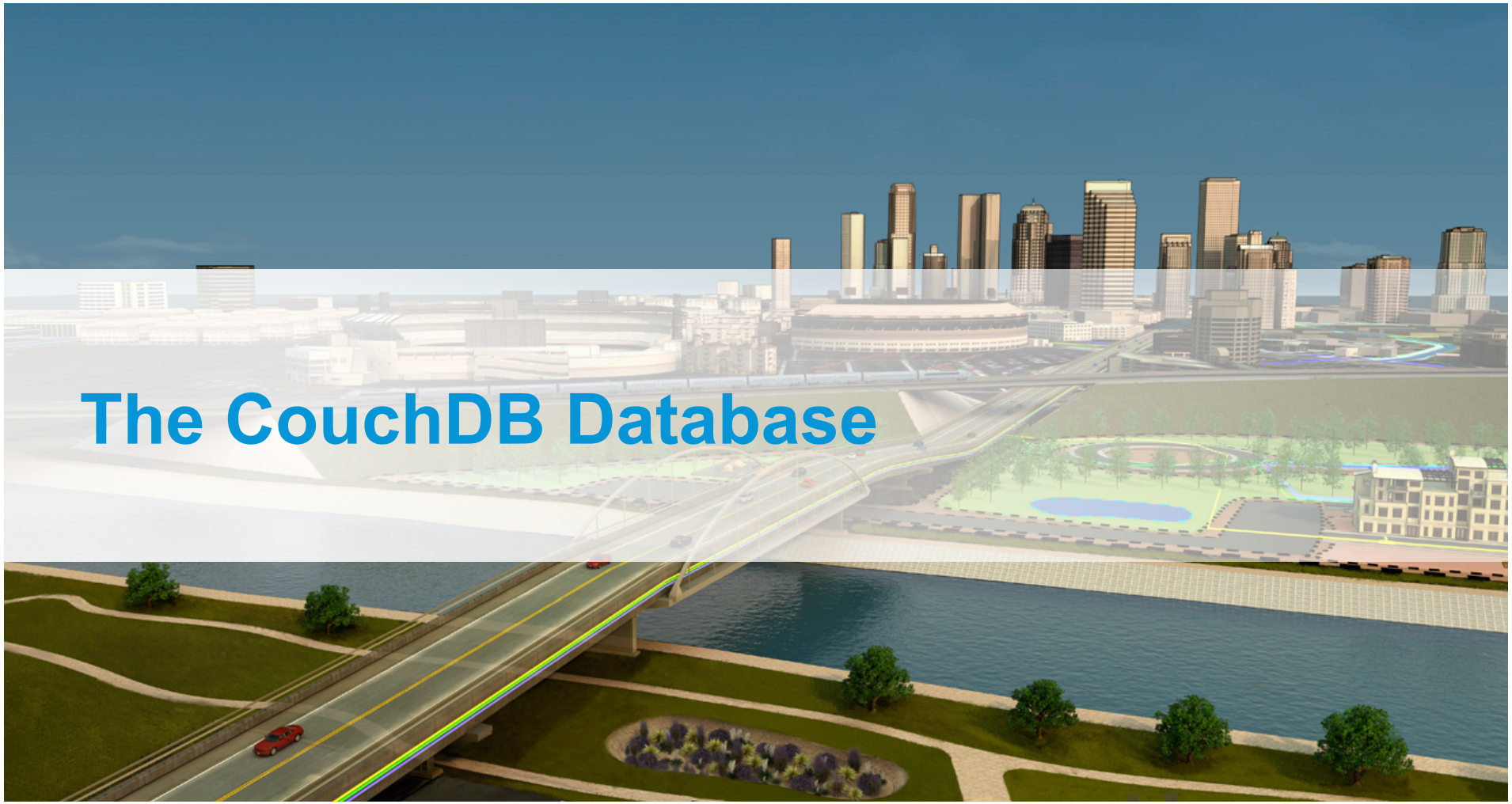
- Note the simple navigation
  - Home page: list all models and select one
  - Model selected: list all levels and select one
  - Level selected: list all rooms and select one
  - Room selected:
    - Display room, furniture, symbol graphics
    - Attach event handlers and enable editor
    - Click and drag to modify family instance transformations
  - Save: update database
- Note the RESTful URLs in the address bar
  - The database interaction is RESTful as well

Free and Simple

- 100% open source components
- 200 hours research
- < 8 hours to rebuild
  - Install components
  - Re-implement from scratch
  - One single file handles all
    - index.html

https://github.com/jeremytammik/roomedit





The CouchDB Database

NoSQL

- Next generation database paradigm
- Address some of the points: non-relational, distributed, open-source, horizontally scalable
- Began 2009 and growing fast
- Frequent other characteristics: schema-free, easy replication support, simple API, eventually consistent / BASE (not ACID), huge amounts of data
- “Not only SQL”
  - http://nosql-database.org, https://en.wikipedia.org/wiki/NoSQL

ACID versus Base

- ACID
  - Atomicity, Consistency, Isolation, Durability
  - Traditional computer science
  - Guarantee that database transactions are processed reliably
- BASE
  - Basic Availability, Soft-state, Eventual consistency
  - Not guaranteed to be in a consistent state at a given moment
  - Consistency is guaranteed, eventually

CouchDB Database Implementation

- Everything is a document
- All documents are JSON
- Every document has built-in id and revision
- The database design is also a document
- The design defines views and attachments

CouchDB Database Interaction

- Relax!
- All interactions are REST
- Management console is Futon
  http://127.0.0.1:5984/_utils/
- Access all documents
  http://127.0.0.1:5984/roomedit/_all_docs
- Include full doc data, not just id and revision
  http://127.0.0.1:5984/roomedit/_all_docs?include_docs=true

BIM Model

- Model is an RVT project file
- Level
- Room
- FamilyInstance represents furniture or equipment
- FamilySymbol defines geometry

BIM Object Relationships and Graphics

- Room has boundary loops and can contain holes
- FamilySymbol has a single boundary loop
- FamilyInstance has a 2D placement
  - Translation
  - Rotation

- Family instance → room → level → model
- Family instance → symbol

NoSQL Database Structure

- DbObj base class
- DbModel
- DbLevel
- DbRoom
- DbFurniture
- DbSymbol

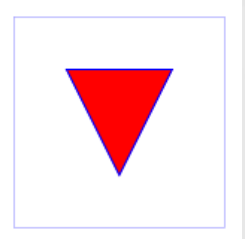
Database Object Relationships

- DbFurniture.symbolId → DbSymbol
- DbFurniture.roomId → DbRoom
- DbRoom.levelId → DbLevel
- DbLevel.modelId → DbModel

Database Object Graphics and Placement

- All graphics represented by SVG path element data

```
<svg width="4cm" height="4cm" viewBox="0 0 400 400"
     xmlns="http://www.w3.org/2000/svg" version="1.1">
  <rect x="1" y="1" width="398" height="398"
        fill="none" stroke="blue" />
  <path d="M 100 100 L 300 100 L 200 300 z"
        fill="red" stroke="blue" stroke-width="3" />
</svg>
```



JSON Symbol Database Document

- Family symbol
- Define geometry


```
{
   "_id": "11cc6e52-519e-49b2-9813-c9561b59a1fd-0005f5fc",
   "_rev": "1-d575ca095533db4ccbed9f7ab2607a12",
   "loop": "M-191 922 L190 922 216 862 190 859 -191 859 -216 862 -216 919Z",
   "type": "symbol",
   "description": "FamilySymbol Furniture <390652 Table ronde a chaises>",
   "name": "Table ronde avec chaises - 01"
}
```

JSON Instance Database Document

- Furniture doc represents family instance and defines
- Relationship to room and family symbol
- Placement = transform = translation + rotation


```
{
   "_id": "11cc6e52-519e-49b2-9813-c9561b59a1fd-0005f65b",
   "_rev": "1-c1b4fc969181267b55dab4c6857fc5d7",
   "roomId": "cbe571b0-0593-4350-a8e6-abf3c9239325-00061210",
   "symbolId": "11cc6e52-519e-49b2-9813-c9561b59a1fd-0005fe6d",
   "transform": "R-90T-10429,1020",
   "type": "furniture",
   "description": "FamilyInstance Furniture <390747 Canapé 3 places>",
   "name": "Canapé 3 places"
}
```

CouchDB Views

- A view defines a map and optional reduce function
- The map produces key-value pairs
- Reduce produces an accumulation

```
exports.models = {
  map: function (doc) {
    if( 'model' == doc.type ) {
      emit(doc, null);
    }
  },
  reduce: function (key, values, rereduce) {
    return sum(values);
  }
}
```

Room Editor Views

- models
- levels
- rooms
- furniture
- symbols
- map_room_to_furniture
- map_level_to_room
- map_model_to_level

```
rooms = {
  map: function (doc) {
    if( 'room' == doc.type ) {
      emit(doc, null);
    }
  }
};
```

```
map_level_to_room = {
  map: function (doc) {
    if( 'room' == doc.type ) {
      emit(doc.levelId, doc);
    }
  }
};
```

Accessing CouchDB Documents and Views

- Specific document

  http://127.0.0.1:5984/roomedit/11cc6e52-519e-49b2-9813-c9561b59a1fd-0005f5fc

- Specific view

  http://127.0.0.1:5984/roomedit/_design/roomedit/_view/models

- Specific key in view

  http://127.0.0.1:5984/roomedit/_design/roomedit/_view/map_level_to_room?key=%22933c4a06-93b8-11d3-80f8-00c04f8efc32-0000001e%22

Entire CouchDB Application Definition

- Kanso loads CouchDB

```
roomedit/data/room_model_9.json
roomedit/index.html
roomedit/kanso.json
roomedit/lib/app.js
roomedit/lib/views.js
roomedit/raphael-min-jt.js
```

- kanso.json

```
{
  "name": "roomedit",
  "attachments": ["index.html",
    "raphael-min-jt.js"],
  "modules": ["lib"],
  "load": "lib/app",
  "dependencies": {
    "attachments": null,
    "modules": null,
    "properties": null,
    "db": null,
    "jquery": null
  }
}
```

Index.html

- HTML scaffolding
- JavaScript query section
- Raphaël SVG generation and interaction
- RESTful database updates
- Called with
  - No argument: list all models
  - Model id: list all levels in model
  - Level id: list all rooms on level
  - Room id: display graphical editor and enable database updates

Minimal Predefined HTML Scaffolding

```
<h1>Room Editor</h1>

<div id="content"></div>

<ul id="navigatorlist"></ul>

<div id="editor"></div>

<p id="current_furniture"></p>

<script type="text/javascript" src="modules.js"></script>
<script type="text/javascript" src="raphael-min-jt.js"></script>
```

- Populated entirely using JavaScript adding HTML and SVG nodes using jquery, raphael and db for CouchDB queries

List all Models

- No input parameter
- View ‘models’, no key
- Populate navigator list

```
db.getView('roomedit', 'models',
  function (err, data) {
    if (err) {
      return alert(err);
    }
    var n = data.rows.length;
    for (var i = 0; i < n; ++i) {
      var doc = data.rows[i].key;
      var s = url + '?modelid=' + doc._id;
      $('<li/>').append($('<a>')
        .attr('href',s)
        .text(doc.name))
      .appendTo('#navigatorlist');
    }
    var p = $('<p/>').appendTo('#content');
    p.append( $('<a/>').text('Home').attr('href',url) );
    p.append( document.createTextNode( three_spaces ) );
    p.append( $('<a/>').text('Back').attr('href',url) );

    $('<p/>')
      .text( 'Please select a model in the list below.' )
      .appendTo('#content');

    $('<p/>')
      .text(n.toString() + ' model’ + pluralSuffix( n ) + dotOrColon( n ) )
      .appendTo('#content');
  }
);
```

List all Levels in Selected Model

- Input parameter ‘modelId’
- Get model document itself
  - Display current selection
- View ‘map_model_to_level’ with model id key
  - Populate navigator list

List all Rooms on Selected Level

- Input parameter ‘levelId’
- Get level document itself → model id
- Get model document
  - Display current selection
- View ‘map_level_to_room’ with level id key
  - Populate navigator list

Display and Edit a Selected Room

- Input parameter ‘roomId’
- Get room document itself → level id
- Get level document → model id
- Get model document
  - Display current selection
- View ‘map_room_to_furniture’ with key room id
- Retrieve family instances → symbol ids
- View ‘symbols’ with list of symbol keys
  - Populate and display SVG editor

Database Query Callback Function

- **Nested** callback functions
- Asynchronous interaction rerouted to a synchronous interface, so to speak

SVG Room Editor

- Using jquery and Raphaël
- Input room + furniture populated with symbol SVG path
- Determine size and aspect ratio
- Instantiate paper = canvas
- Place room and attach tooltip events mouseover + out
- Place furniture and attach identification, drag and tooltip events

Challenges

- Many!
- NoSQL, CouchDB, views, CouchApp and Kanso
- JavaScript, SVG, event handling, mobile devices
- Mapping to negative Y axis and restore for update
- Handle nested database callback functions
- Retrieving database changes
- Subscribe to continuous changes?
- This is a simple learning sample
- Infinite possible real-life applications

Conclusion

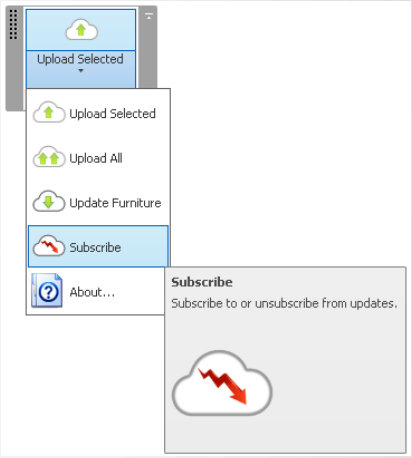
- NoSQL is great!
- Open source is free and effective
- REST and JSON is powerful and simple to work with
- Server-side scripting pays off
  - Avoid programming on the mobile device itself
- KISS!

RoomEditorApp – The Revit Add-In

RoomEditorApp Features

- External application to present the user interface
- Idling challenges and benchmarking timer
- Determine rooms and furniture boundary loops
- Transform Revit graphics to SVG path descriptions
- Temporary graphical display of boundary loops
- Represent the database model
- Interact with the cloud database
- Utilities for unit conversion, formatting, messages, folder browser, etc.
- Let's look at the code together in the debugger...

  https://github.com/jeremytammik/RoomEditorApp



Revit Add-In Access to CouchDB

- How does the Revit add-in access CouchDB?
  - Upload model data
  - Retrieve database changes
  - Subscribe to changes
- DreamSeat

  https://github.com/vdaron/DreamSeat
  - Easy!
  - Good job!

Room Editor Documentation and Materials

- The Building Coder Revit API blog categories
  - http://thebuildingcoder.typepad.com
    - Cloud
    - Desktop
    - Mobile
- Jeremy's GitHub repositories
  - https://github.com/jeremytammik
    - Roomedit
- Room editor playground
  - http://jt.iriscouch.com/roomedit/_design/roomedit/index.html

Learning More

Revit Developer Center: DevTV and My First Plugin Introductions, SDK, API Help, Samples

http://www.autodesk.com/developrevit

Developer Guide and Online Help

http://www.autodesk.com/revitapi-wikihelp

Revit API Trainings, Webcasts and Archives

http://www.autodesk.com/apitraining > Revit API

Discussion Group

http://discussion.autodesk.com > Revit Architecture > Revit API

API Training Classes

http://www.autodesk.com/apitraining

ADN AEC DevBlog

http://adndevblog.typepad.com/aec

The Building Coder, Jeremy Tammik's Revit API Blog

http://thebuildingcoder.typepad.com

ADN, The Autodesk Developer Network

http://www.autodesk.com/joinadn and http://www.autodesk.com/adnopen

DevHelp Online for ADN members

http://adn.autodesk.com