NoSQL

|  |  |
| --- | --- |
| * Scaling – Spread over multiple machines | * Eventual Consistency – Not immediately updated |

NoSQL Categories

|  |  |
| --- | --- |
| * Key/Value * Document | * Column * Graph |

Firebase Essentials

|  |  |
| --- | --- |
| * Key/Value – Looks like a document db * Hosted – Low Maintenance * URL-oriented – Unique URL for each piece of data | * Real-time – Fast & Convenient * Integrated Authentication – Several Options |

|  |  |
| --- | --- |
| Firebase Architectural Scenario #1 | Firebase Architectural Scenario #2 |
|  |  |
| Firebase Architectural Scenario #3 | Firebase Architectural Scenario #3 |
|  |  |

Understanding Data – Refs

**Creating refs**

var url = <https://example.firebaseio.com/>;

var ref = new Firebase(url);

**Creating multiple refs**

var url = <https://example.firebaseio.com/>;

var baseRef = new Firebase(url);

var hobbitRef = new Firebase(url+”books/the\_hobbit/”);

**The parent() method**

var url = <https://example.firebaseio.com/>;

var starWarsRef = new Firebase(url+”movies/star\_wars/”);

var moviesRef = starWarsRef.parent();

**The root() method**

var url = <https://example.firebaseio.com/>;

var starWarsRef = new Firebase(url+”movies/star\_wars/”);

var moviesRef = starWarsRef.root();

|  |  |
| --- | --- |
| Data Design Principles | * Don’t Treat Firebase like a Relational Database * Use Root Branches as Your Primary Container |