

Derby

realtime apps
real easy

Team

Brian Noguchi

Nate Smith

Joseph Gentle

Our approach

Demos

hello.js

```
var hello = require('derby').createApp(module);

hello.view.make('Body',
  'Holler: <input value="(message)"><h1>(message)</h1>'
);

hello.get('/', function(page, model) {
  model.subscribe('message', function() {
    page.render();
  });
});
```

hello.js

```
var hello = require('derby').createApp(module);

hello.view.make('Body',
  'Holler: <input value="(message)"><h1>(message)</h1>'
);

hello.get('/', function(page, model) {
  model.subscribe('message', function() {
    page.render();
  });
});
```

hello.js

```
var hello = require('derby').createApp(module);

hello.view.make('Body',
  'Holler: <input value="(message)"><h1>(message)</h1>'
);

hello.get('/', function(page, model) {
  model.subscribe('message', function() {
    page.render();
  });
});
```

hello.js

```
var hello = require('derby').createApp(module);

hello.view.make('Body',
  'Holler: <input value="( (message)) "><h1>( (message))</h1>'
);

hello.get('/', function(page, model) {
  model.subscribe('message', function() {
    page.render();
  });
});
```


hello.js

```
var hello = require('derby').createApp(module);

hello.view.make('Body',
  'Holler: <input value="( (message)) "><h1>( (message))</h1>'
);

hello.get('/', function(page, model) {
  model.subscribe('message', function() {
    page.render();
  });
});
```

hello.js

```
var hello = require('derby').createApp(module);

hello.view.make('Body',
  'Holler: <input value="(message)"><h1>(message)</h1>'
);

hello.get('/', function(page, model) {
  model.subscribe('message', function() {
    page.render();
  });
});
```

hello.js

```
var hello = require('derby').createApp(module);

hello.view.make('Body',
  'Holler: <input value="( (message)) "><h1>( (message))</h1>'
);

hello.get('/', function(page, model) {
  model.subscribe('message', function() {
    page.render();
  });
});
```

server.js

```
var express = require('express'),  
    hello = require('./hello'),  
    server = express.createServer()  
        .use(express.static(__dirname + '/public'))  
        .use(hello.router()),  
  
    store = hello.createStore({ listen: server });  
  
server.listen(3000);
```

All app code *client and server*

Realtime synced

- Responsive
- No glue code
- Offline
- SEO friendly

Write views first

<Body:>

<h1>{{_stock.company}}: {{_stock.symbol}}**</h1>**

<p id="price">((_stock.price))**</p>**

{{#_stock.delay}}

This price may be delayed up to {{_stock.delay}}.

{{^}}

This price is updated in realtime.

{{/}}

<h2>Related**</h2>**

<table>

((#_stock.related))

<tr>((.stats > stat))**</tr>**

((/))

</table>

<stat:>

<td class="((.color)) ((.align))**">**((.value))**</td>**

<Body:>

<h1>{{_stock.company}}: {{_stock.symbol}}**</h1>**

<p id="price">((_stock.price))**</p>**

{{#_stock.delay}}

This price may be delayed up to {{_stock.delay}}.

{{^}}

This price is updated in realtime.

{{/}}

<h2>Related**</h2>**

<table>

((#_stock.related))

<tr>((.stats > stat))**</tr>**

((/))

</table>

<stat:>

<td class="((.color)) **((.align))"**>((.value))**</td>**

<Body:>

<h1>{{_stock.company}}: {{_stock.symbol}}**</h1>**

<p id="price">({_stock.price})**</p>**

{{#_stock.delay}}

This price may be delayed up to {{_stock.delay}}.

{{^}}

This price is updated in realtime.

{{/}}

<h2>Related**</h2>**

<table>

((#_stock.related))

<tr>((.stats > stat))**</tr>**

((/))

</table>

<stat:>

<td class="((.color)) ((.align))" > ((.value))**</td>**

```

<Body:>
  <h1>{{_stock.company}}: {{_stock.symbol}}</h1>
  <p id="price">({_stock.price})</p>

  {{#_stock.delay}}
    This price may be delayed up to {{_stock.delay}}.
  {{^}}
    This price is updated in realtime.
  {{/}}

  <h2>Related</h2>
  <table>
    ( (#_stock.related) )
    <tr>({_stock.related.stats > stat})</tr>
    ( (/) )
  </table>

  <stat:>
    <td class="({_stock.related.color}) ({_stock.related.align})">({_stock.related.value})</td>

```

<Body:>

<h1>{{_stock.company}}: {{_stock.symbol}}**</h1>**

<p id="price">((_stock.price))**</p>**

{{#_stock.delay}}

This price may be delayed up to {{_stock.delay}}.

{{^}}

This price is updated in realtime.

{{/}}

<h2>Related**</h2>**

<table>

((#_stock.related))

<tr>((.stats > stat))**</tr>**

((/))

</table>

<stat:>

<td class="((.color)) ((.align))**">**((.value))**</td>**

```

<Body:>
  <h1>{{_stock.company}}: {{_stock.symbol}}</h1>
  <p id="price">({_stock.price})</p>

  {{#_stock.delay}}
    This price may be delayed up to {{_stock.delay}}.
  {{^}}
    This price is updated in realtime.
  {{/}}

  <h2>Related</h2>
  <table>
    ({{#_stock.related}}
      <tr>({.stats > stat})</tr>
    (/{/))
  </table>

  <stat:>
    <td class="({.color}) ({.align})">({.value})</td>

```

```
<Body:>
```

```
  <h1>{{_stock.company}}: {{_stock.symbol}}</h1>
```

```
  <p id="price">({_stock.price})</p>
```

```
  {{#_stock.delay}}
```

```
    This price may be delayed up to {{_stock.delay}}.
```

```
  {{^}}
```

```
    This price is updated in realtime.
```

```
  {{/}}
```

```
  <h2>Related</h2>
```

```
  <table>
```

```
    (({_stock.related})
```

```
      <tr>({.stats > stat})</tr>
```

```
    ( (/ ))
```

```
  </table>
```

```
<stat:>
```

```
  <td class="({.color}) ({.align})">({.value})</td>
```

```
<Body:>
```

```
  <h1>{{_stock.company}}: {{_stock.symbol}}</h1>
```

```
  <p id="price">({_stock.price})</p>
```

```
  {{#_stock.delay}}
```

```
    This price may be delayed up to {{_stock.delay}}.
```

```
  {{^}}
```

```
    This price is updated in realtime.
```

```
  {{/}}
```

```
  <h2>Related</h2>
```

```
  <table>
```

```
    ( (#_stock.related) )
```

```
      <tr>(.stats > stat)</tr>
```

```
    ( (/ ) )
```

```
  </table>
```

```
<stat:>
```

```
  <td class="(.color) (.align)">(.value)</td>
```

**Same routes:
client and server**

```
get( '/:room', function(page, model, params) {  
    page.render({title: params.room});  
});
```

```
post( '/cart', function(page, model, params) {  
    var body = params.body;  
    // ...  
    page.redirect( 'back' );  
});
```

```
put( ... );
```

```
del( ... );
```



```
get( '/:room', function(page, model, params) {  
    page.render({title: params.room});  
});
```

```
post( '/cart', function(page, model, params) {  
    var body = params.body;  
    // ...  
    page.redirect( 'back' );  
});
```

```
put( ... );
```

```
del( ... );
```

```
get( '/:room', function(page, model, params) {  
    page.render({title: params.room});  
});
```

```
post( '/cart', function(page, model, params) {  
    var body = params.body;  
    // ...  
    page.redirect( 'back' );  
});
```

```
put( ... );
```

```
del( ... );
```

```
get( '/:room', function(page, model, params) {  
    page.render({title: params.room});  
});
```

```
post( '/cart', function(page, model, params) {  
    var body = params.body;  
    // ...  
    page.redirect( 'back' );  
});
```

```
put( ... );
```

```
del( ... );
```

```
get( '/:room', function(page, model, params) {  
    page.render({title: params.room});  
});
```

```
post( '/cart', function(page, model, params) {  
    var body = params.body;  
    // ...  
    page.redirect( 'back' );  
});
```

```
put( ... );
```

```
del( ... );
```

Data syncing

```
<Body:>
```

```
( (_room.name) )
```

```
<button x-bind="click:shout">Shout it!</button>
```

```
get( '/:room', function(page, model, params) {
```

```
  var room = params.room;
```

```
  model.subscribe( 'rooms.' + room, function() {
```

```
    model.set( '_room', model.ref( 'rooms.' + room) );
```

```
    model.set( '_room.name', room);
```

```
    page.render();
```

```
  } );
```

```
});
```

```
ready( function(model) {
```

```
  exports.shout = function() {
```

```
    var name = model.get( '_room.name' ).toUpperCase();
```

```
    model.set( '_room.name', name);
```

```
  };
```

```
});
```

```
<Body:>
```

```
( (_room.name) )
```

```
<button x-bind="click:shout">Shout it!</button>
```

```
get( '/:room', function(page, model, params) {
```

```
  var room = params.room;
```

```
  model.subscribe('rooms.' + room, function() {
```

```
    model.set('_room', model.ref('rooms.' + room));
```

```
    model.set('_room.name', room);
```

```
    page.render();
```

```
  });
```

```
});
```

```
ready(function(model) {
```

```
  exports.shout = function() {
```

```
    var name = model.get('_room.name').toUpperCase();
```

```
    model.set('_room.name', name);
```

```
  };
```

```
});
```

```
<Body:>
```

```
( (_room.name) )
```

```
<button x-bind="click:shout">Shout it!</button>
```

```
get( '/:room', function(page, model, params) {
```

```
  var room = params.room;
```

```
  model.subscribe( 'rooms.' + room, function() {
```

```
    model.set( '_room', model.ref( 'rooms.' + room) );
```

```
    model.set( '_room.name', room);
```

```
    page.render();
```

```
  });
```

```
});
```

```
ready(function(model) {
```

```
  exports.shout = function() {
```

```
    var name = model.get( '_room.name' ).toUpperCase();
```

```
    model.set( '_room.name', name);
```

```
  };
```

```
});
```



```
<Body:>
```

```
  ( (_room.name) )
```

```
  <button x-bind="click:shout">Shout it!</button>
```

```
get( '/:room', function(page, model, params) {
```

```
  var room = params.room;
```

```
  model.subscribe( 'rooms.' + room, function() {
```

```
    model.set( '__room', model.ref( 'rooms.' + room) );
```

```
    model.set( '__room.name', room);
```

```
    page.render();
```

```
  } );
```

```
});
```

```
ready( function(model) {
```

```
  exports.shout = function() {
```

```
    var name = model.get( '__room.name' ).toUpperCase();
```

```
    model.set( '__room.name', name);
```

```
  };
```

```
});
```

```
<Body:>
```

```
( (_room.name) )
```

```
<button x-bind="click:shout">Shout it!</button>
```

```
get( '/:room', function(page, model, params) {
```

```
  var room = params.room;
```

```
  model.subscribe( 'rooms.' + room, function() {
```

```
    model.set( '_room', model.ref( 'rooms.' + room) );
```

```
    model.set( '_room.name', room);
```

```
    page.render();
```

```
  } );
```

```
});
```

```
ready( function(model) {
```

```
  exports.shout = function() {
```

```
    var name = model.get( '_room.name' ).toUpperCase();
```

```
    model.set( '_room.name', name);
```

```
  };
```

```
});
```

```
<Body:>
```

```
( (_room.name) )
```

```
<button x-bind="click:shout">Shout it!</button>
```

```
get( '/:room', function(page, model, params) {
```

```
  var room = params.room;
```

```
  model.subscribe( 'rooms.' + room, function() {
```

```
    model.set( '_room', model.ref( 'rooms.' + room) );
```

```
    model.set( '_room.name', room);
```

```
    page.render();
```

```
  });
```

```
});
```

```
ready(function(model) {
```

```
  exports.shout = function() {
```

```
    var name = model.get( '_room.name' ).toUpperCase();
```

```
    model.set( '_room.name', name);
```

```
  };
```

```
});
```

```
<Body:>
```

```
( (_room.name) )
```

```
<button x-bind="click:shout">Shout it!</button>
```

```
get( '/:room', function(page, model, params) {
```

```
  var room = params.room;
```

```
  model.subscribe( 'rooms.' + room, function() {
```

```
    model.set( '_room', model.ref( 'rooms.' + room) );
```

```
    model.set( '_room.name', room);
```

```
    page.render();
```

```
  });
```

```
});
```

```
ready(function(model) {
```

```
  exports.shout = function() {
```

```
    var name = model.get( '_room.name' ).toUpperCase();
```

```
    model.set( '_room.name', name);
```

```
  };
```

```
});
```

```
<Body:>
```

```
( (_room.name) )
```

```
<button x-bind="click:shout">Shout it!</button>
```

```
get( '/:room', function(page, model, params) {
```

```
  var room = params.room;
```

```
  model.subscribe( 'rooms.' + room, function() {
```

```
    model.set( '_room', model.ref( 'rooms.' + room) );
```

```
    model.set( '_room.name', room);
```

```
    page.render();
```

```
  });
```

```
});
```

```
ready(function(model) {
```

```
  exports.shout = function() {
```

```
    var name = model.get( '_room.name' ).toUpperCase();
```

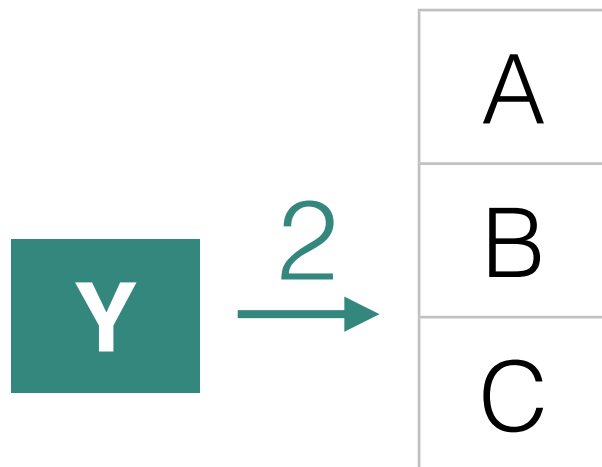
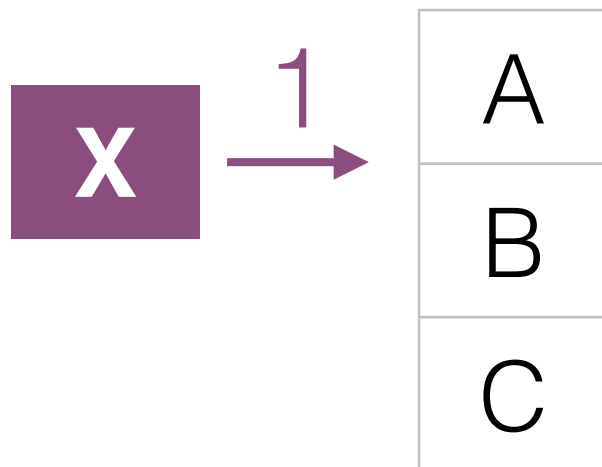
```
    model.set( '_room.name', name);
```

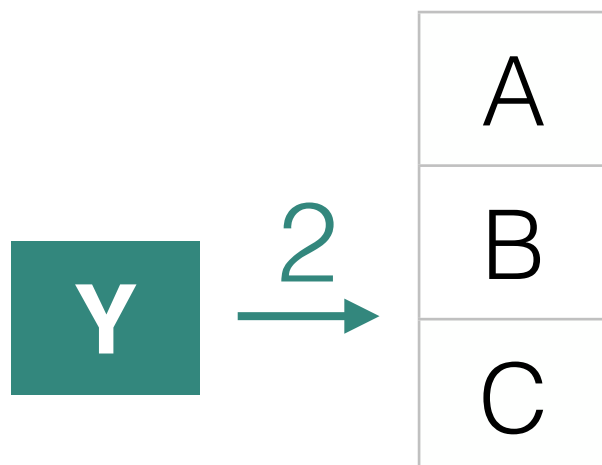
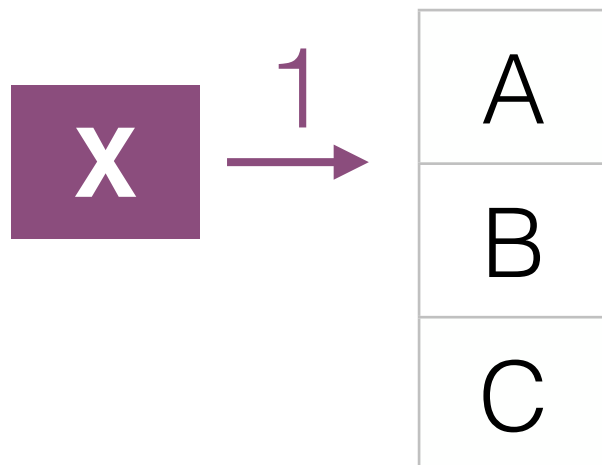
```
  };
```

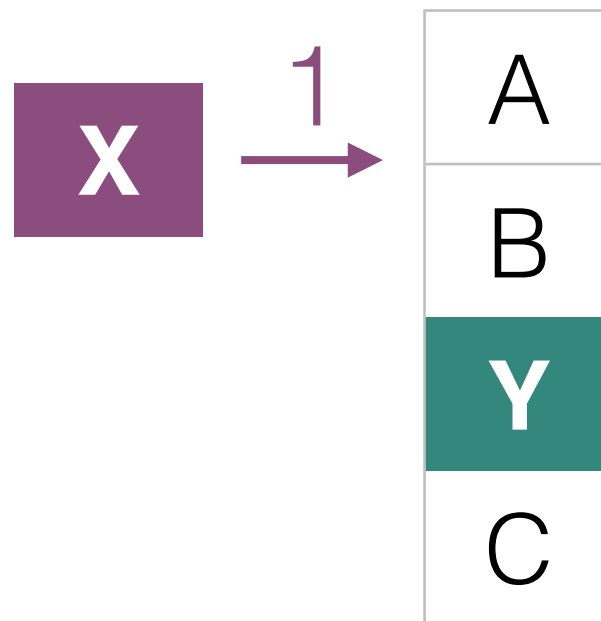
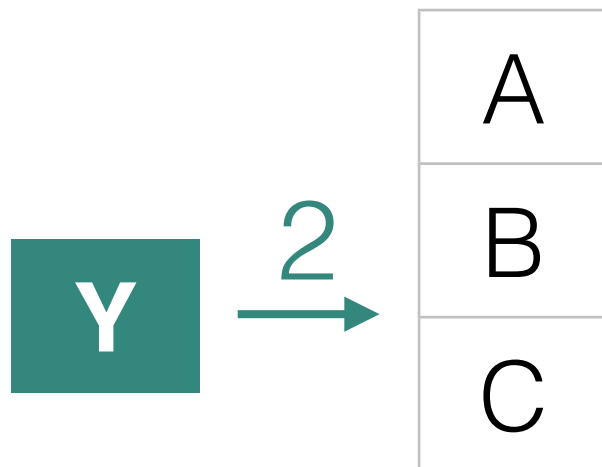
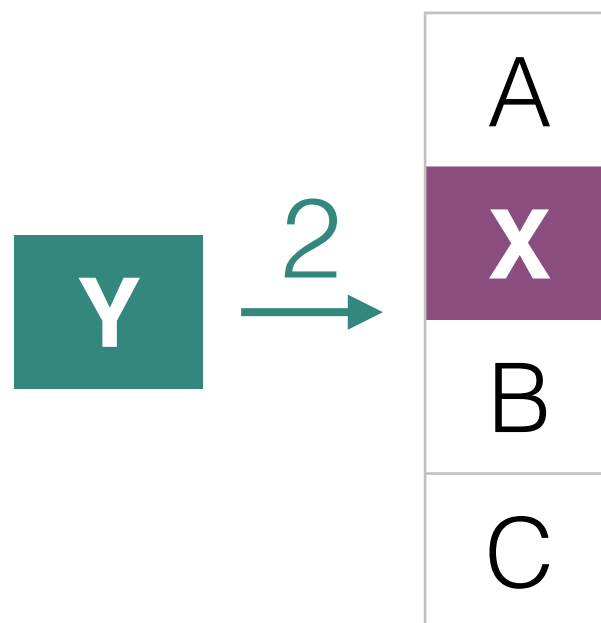
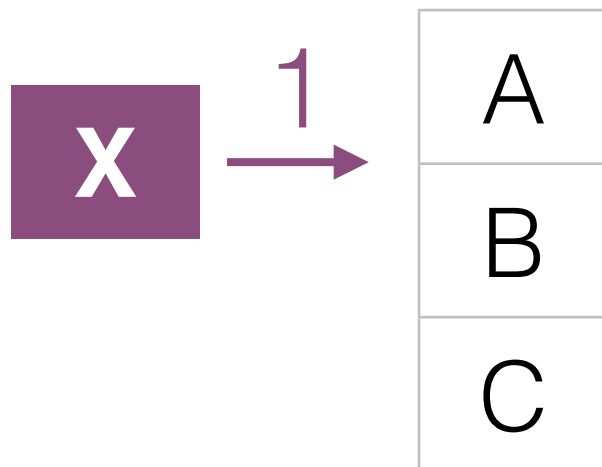
```
});
```

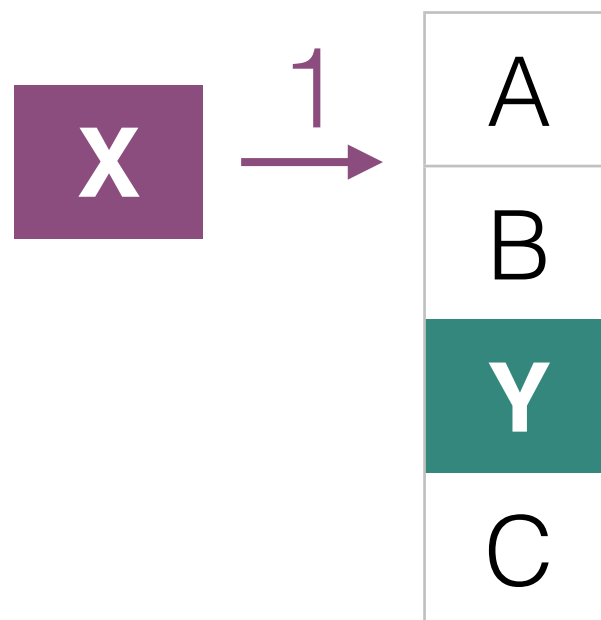
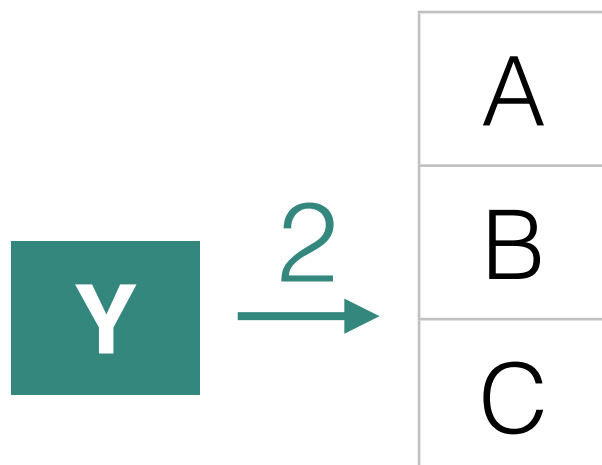
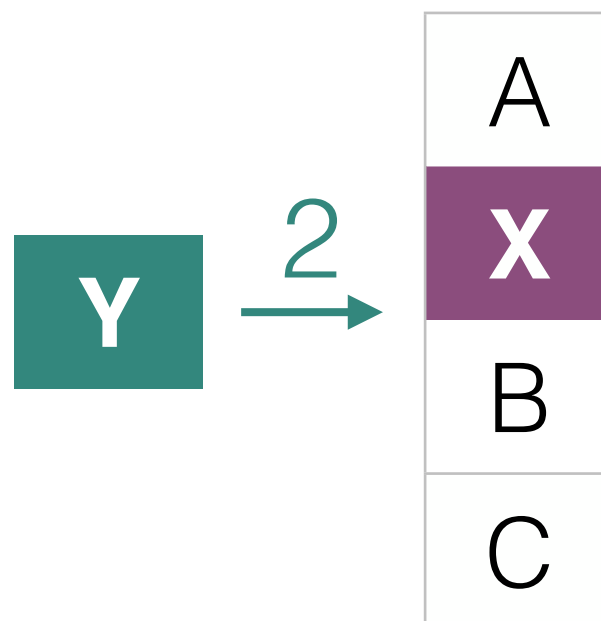
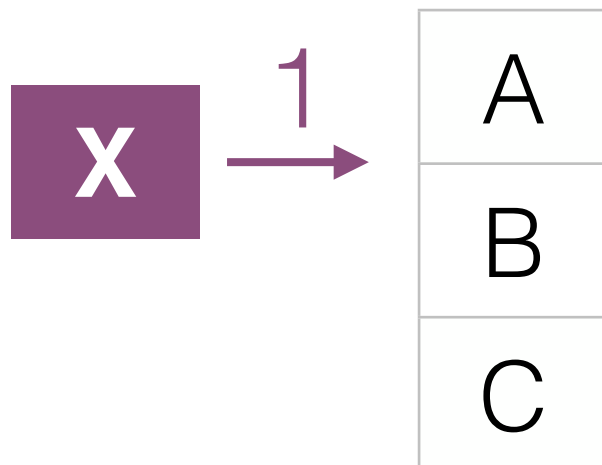
Conflict resolution

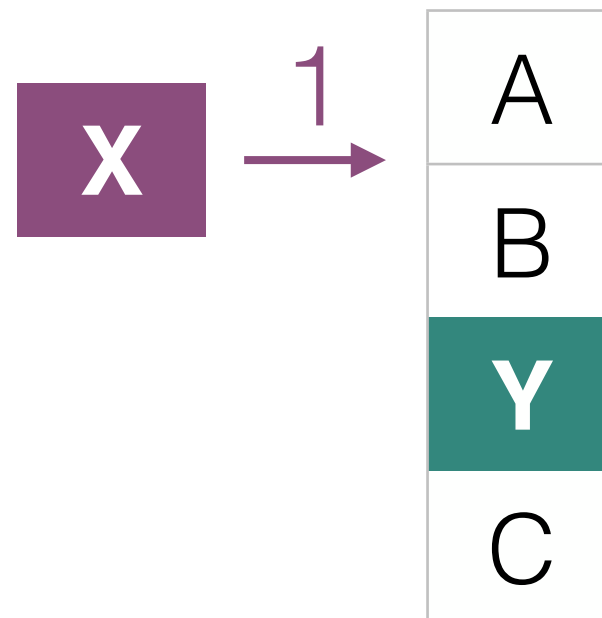
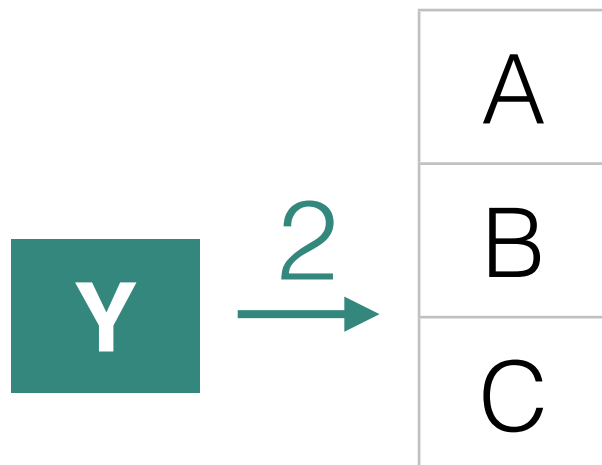
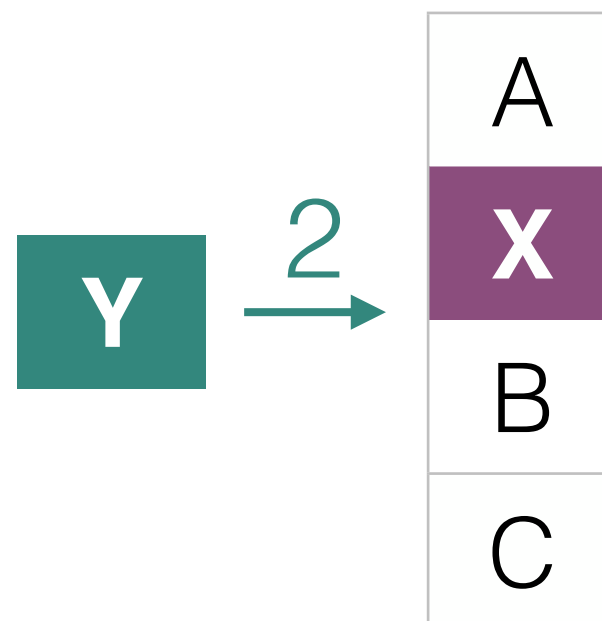
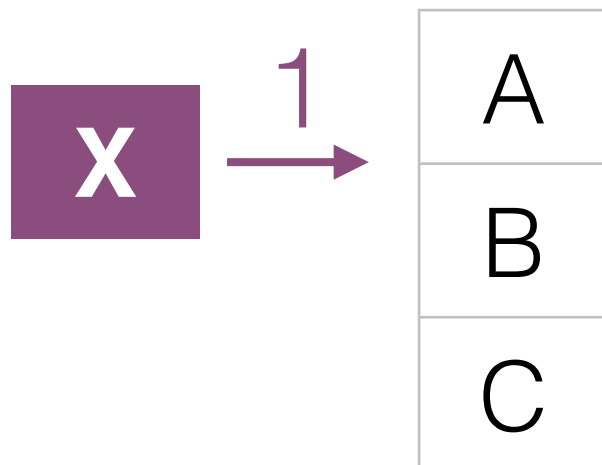
Say what?











```
model.insertAfter('list', 1, 'X', function(err, value) {  
    // err == null  
});
```

```
model.insertAfter('list', 2, 'Y', function(err, value) {  
    // err == 'conflict'  
});
```

Offline

Letters

Database storage


```
var User = Schema.extend( 'User', {  
  _id: String,  
  age: Number  
});
```

```
User.createDataSchema(mongo, {  
  _id: ObjectId,  
  age: Number  
});
```

```
var User = Schema.extend('User', {  
  _id: String,  
  age: Number  
});
```

```
User.createDataSchema(mongo, {  
  _id: ObjectId,  
  age: Number  
});
```

```
var User = Schema.extend( 'User', {  
  _id:      String,  
  newsfeed: [FeedItem],  
  transactions: [Transaction]  
});
```

```
User.createDataSchema(redis, {}, {  
  newsfeed:      redis.FeedItem.find({ userId: '@user._id' })  
});
```

```
User.createDataSchema(mysql, {}, {  
  transactions: mysql.Transaction.find({ userId: '@user._id' })  
});
```

```
var User = Schema.extend('User', {  
  _id:      String,  
  newsfeed: [FeedItem],  
  transactions: [Transaction]  
});
```

```
User.createDataSchema(redis, {}, {  
  newsfeed:      redis.FeedItem.find({ userId: '@user._id' })  
});
```

```
User.createDataSchema(mysql, {}, {  
  transactions: mysql.Transaction.find({ userId: '@user._id' })  
});
```

All app code *client and server*

Realtime synced

- Responsive
- No glue code
- Offline
- SEO friendly

EtherPad

Really real-time collaboration.

8 lines

pad.js

```
var pad = require('derby').createApp(module);

pad.view.make('Body', '<textarea>(_pad.text)</textarea>');

pad.get('/:pad', function(page, model, params) {
  model.subscribe({ _pad: 'pads.' + params.pad }, function() {
    model.setNull('_pad.text', model.ot('Edit here...'));
    page.render();
  });
});
```


Derby

realtime apps
real easy

<http://derbyjs.com>
[@derbyjs](#)