AHK: **[?idi]** (injector dot instantiate)  
SOURCE: http://stackoverflow.com/questions/31367878/   
$injector.instantiate(Type, [locals]);  
Creates a new instance of the given Type. Takes a constructor function, then invokes the new instance with the arguments specified in the constructor annotation.

|  |
| --- |
| Assume the following 'class': |
| function Person (fName, lName, $http, $q) {  return {  first\_name: fName,  last\_name: lName,  http: $http,  q: $q  }  } |

|  |
| --- |
| Now, if we wanted to create a new Person in our controller, we could do it like this: |
| app.controller('...', function ($injector) {  var $http = $injector.get('$http');  var $q = $injector.get('$q');  var p = new Person('kasper', 'lewau', $http, $q);  console.log(p); // { first\_name: 'kasper', last\_name: 'lewau', http: $http, q: $q };  }); |

|  |
| --- |
| magine Person had ~20 or so dependencies, and we were fetching each and every one of them with the $injector.get method.  Cumbersome! And - you would need to keep your parameters & arguments in sync. ugh.  Instead, you can do this: |
| app.controller('...', function ($injector) {  var p = $injector.instantiate(Person, {  fName: 'kasper',  lName: 'lewau'  });  console.log(p); // { first\_name: 'kasper', last\_name: 'lewau', http: $http, q: $q };  }); |

|  |
| --- |
| And - if we wanted to, we could supply locals to the .instantiate call, so as to override what the internal $injector.get() would normally get when instantiating. |
| var p = $injector.instantiate(Person, {  fName: 'kasper',  lName: 'lewau'  },{  $http: 'Nothing!', $q: 'Nothing!'  });  console.log(p);  // { first\_name: 'kasper', last\_name: 'lewau', http: 'Nothing!', q: 'Nothing!' }; |