

# Key-Value Coding

# O que é?

Uma maneira alternativa de acessar propriedades...

Com algumas funções para facilitar a nossa vida. (:

Como assim?

...

```
@interface Developer : NSObject
```

```
@property (nonatomic, assign) NSInteger age;
```

```
@property (nonatomic, strong) NSString *city;
```

```
@end
```

...

```
...  
Developer *dev = [[Developer alloc] init];
```

```
//Setters
```

```
dev.city = @"Campinas";
```

```
dev.age = 21;
```

```
//or
```

```
[dev setCity:@"Campinas"];
```

```
[dev setAge:21];
```

```
//Getters
```

```
NSInteger age = dev.age;
```

```
NSString *city = dev.city;
```

```
//or
```

```
NSInteger age = [dev age];
```

```
NSString *city = [dev city];
```

```
...
```

...

```
Developer *dev = [[Developer alloc] init];
```

```
//Setters
```

```
[dev setValue:@"Campinas" forKey:@"city"];
```

```
[dev setValue:@(21) forKey:@"age"];
```

```
//Getters
```

```
NSInteger age = [[dev valueForKey:@"age"] intValue];
```

```
NSString *city = [dev valueForKey:@"city"];
```

...

Mas... como isso pode  
me ajudar?

```
...
/*
 [
  {
    "age" : 21,
    "city" : "Campinas"
  },
  {
    "age" : 26,
    "city" : "Belo Horizonte"
  },
  ...
 ]
*/
...
```



...

```
@[  
  @{  
    @"age" : @(21),  
    @"city" : @"Campinas"  
  },  
  @{  
    @"age" : @(26),  
    @"city" : @"Belo Horizonte"  
  },  
  ...  
]
```

...

...

```
+ (NSArray *)parseJSONArray:(NSArray *)jsonArray
{
    NSMutableArray *entities = [NSMutableArray array];
    for (NSDictionary *dictionary in jsonArray)
    {
        Developer *dev = [[Developer alloc] init];
        dev.city = dictionary[@"city"]; // -unsafe
        dev.age = [dictionary[@"age"] intValue]; // -unsafe
        //... more n lines to more n properties ...
        [entities addObject:dev];
    }
    return [entities copy];
}
```

...

```
...
+ (NSArray *)parseJSONArray:(NSArray *)jsonArray
{
    NSMutableArray *entities = [NSMutableArray array];
    for (NSDictionary *dictionary in jsonArray)
    {
        Developer *dev = [[Developer alloc] init];
        for (NSString *key in [dictionary allKeys])
        {
            [dev setValue:dictionary[key] forKey:key];
        }
        [entities addObject:dev];
    }
    return [entities copy];
}
...
```

...

@implementation Developer

```
- (void)setValue:(id)value forKey:(NSString *)key  
{  
    NSLog(@"Sorry, I don't have any property named %@", key);  
}
```

@end

...

Key

X

KeyPath

...

```
Developer *dev = [[Developer alloc] init];
```

```
dev.city = @"Campinas";
```

```
NSNumber *cityLength = [dev.city valueForKey:@"length"];
```

```
//or
```

```
NSNumber *cityLength = [dev valueForKeyPath:@"city.length"];
```

```
//a bit slower
```

...

# Collections

(Arrays and Sets)

...

//Without KVC

```
for (Developer *dev in developers)
{
    dev.city = @"Campinas";
}
```

//Using KVC

```
[developers setValue:@"Campinas" forKey:@"city"];
```

...



...

//Without KVC

```
NSMutableArray *cityNames = [NSMutableArray array];  
for (Developer *dev in developers)  
{  
    [cityNames addObject:dev.city];  
}
```

//Using KVC

```
NSArray *cityNames = [developers valueForKey:@"city"];
```

//cityNames result:

```
@[  
    @"Campinas",  
    @"Belo Horizonte"  
];
```

...

...

```
//KeyPaths
```

```
NSArray *cityNames = [developers valueForKeyPath:@"city.uppercaseString"];
```

```
//cityNames result:
```

```
@[
```

```
    @"CAMPINAS",
```

```
    @"BELO HORIZONTE"
```

```
];
```

...

# Collection Operators

- @max
- @min
- @avg
- @sum
- @distinctUnionOfObjects

...

```
[object valueForKeyPath:@"<keyPath>"];
```

<keyPath> :

```
@"<keypathToCollection>.<@collectionOperator>.<keypathToProperty>"
```

...

...

```
@interface Company : NSObject
```

```
@property (nonatomic, strong) NSArray *developers;
```

```
@end
```

...

...

```
Company *company = [MySuperCache fetchCompany];
```

```
NSNumber *maxAge = [company valueForKeyPath:@"developers.@max.age"];
```

```
NSNumber *minAge = [company valueForKeyPath:@"developers.@min.age"];
```

```
NSNumber *ageAverage = [company valueForKeyPath:@"developers.@avg.age"];
```

```
NSNumber *ageSum = [company valueForKeyPath:@"developers.@sum.age"];
```

```
NSArray *cities = [company valueForKeyPath:@"developers.@distinctUnionOfObjects.city"];
```

...

...

```
//Average Without KVC
```

```
NSInteger sum = 0;
```

```
for (Developer *dev in company.developers)
```

```
{
```

```
    sum += dev.age;
```

```
}
```

```
NSInteger count = company.developers.count;
```

```
CGFloat average = count ? sum / count : 0;
```

```
NSNumber *ageAverage = @(average);
```

...

# Referências

- <https://developer.apple.com/library/mac/#documentation/cocoa/Conceptual/KeyValueCoding/Articles/KeyValueCoding.html>
- <http://bou.io/KVCCustomOperators.html>
- <http://www.amazon.com/iOS-Programming-Pushing-Limits-Application/dp/1118449959>



Obrigado. (-: