
From: Larry Visoski [REDACTED]
Sent: Sunday, March 23, 2014 2:42 AM
To: Karyna Shuliak; [REDACTED]
Cc: Lesley Groff
Subject: Fwd: Oakland arrival

```
/** * Recursively merges own enumerable properties of the source object(s), that * don't resolve to `undefined` into the
destination object. Subsequent sources * overwrite property assignments of previous sources. If `customizer` is *
provided it is invoked to produce the merged values of the destination and * source properties. If `customizer` returns
`undefined` merging is handled * by the method instead. The `customizer` is bound to `thisArg` and invoked * with five
arguments: (objectValue, sourceValue, key, object, source). * * @static * @memberOf _ * @category Object * @param
{Object} object The destination object. * @param {...Object} [sources] The source objects. * @param {Function}
[customizer] The function to customize assigned values. * @param {*} [thisArg] The `this` binding of `customizer`. *
@returns {Object} Returns `object`. * @example * * var users = { * 'data': [{ 'user': 'barney' }, { 'user': 'fred' }] * }; *
var ages = { * 'data': [{ 'age': 36 }, { 'age': 40 }] * }; * * _.merge(users, ages); * // => { 'data': [{ 'user': 'barney', 'age': 36 }, {
'user': 'fred', 'age': 40 }] } * * // using a customizer callback * var object = { * 'fruits': ['apple'], * 'vegetables': ['beet'] * }; *
var other = { * 'fruits': ['banana'], * 'vegetables': ['carrot'] * }; * * _.merge(object, other, function(a, b) { * if
(_.isArray(a)) { * return a.concat(b); * } * }); * // => { 'fruits': ['apple', 'banana'], 'vegetables': ['beet', 'carrot'] } */
var merge = createAssigner(baseMerge); module.exports = merge;
```