<https://docs.angularjs.org/api/ng/service/$q>

oauthFixInterception

angular.module('app.services')  
 .service('oauthFixInterceptor', ['$q', '$rootScope', 'OAuthToken',  
 **function** ($q, $rootScope, OAuthToken) {  
 **return** {  
 request: **function** (config) {  
 config.headers = config.headers || {};  
 **if** (OAuthToken.getAuthorizationHeader()) {  
 config.headers = config.headers || {};  
 config.headers.Authorization = OAuthToken.getAuthorizationHeader();  
 }  
 **return** config;  
 },  
 responseError: **function** (rejection) {  
 //peguei um objeto que vamos usar para ptrotelar a decisao sobre resposta e erro  
 //  
 **var** deferred = $q.defer();  
 **if** (400 === rejection.status && rejection.data &&  
 ("invalid\_request" === rejection.data.error ||  
 "invalid\_grant" === rejection.data.error)) {  
 OAuthToken.removeToken();  
 $rootScope.$emit("oauth:error", {rejection: rejection, deferred: deferred});  
 }  
 **if** (401 === rejection.status && rejection.data &&  
 "access\_denied" === rejection.data.error ||  
 rejection.headers("www-authenticate") && 0 ===  
 rejection.headers("www-authenticate").indexOf("Bearer")) {  
 $rootScope.$emit("oauth:error", {rejection: rejection, deferred: deferred});  
 //retornando a promessa  
 **return** deferred.promise;  
 }  
 **return** $q.reject(rejection);  
  
 }  
 };  
 }]);

App.js

// Refresh token when a `invalid\_token` error occurs.  
**if** ('access\_denied' === data.rejection.data.error) {  
 **return** OAuth.getRefreshToken().then(**function**(data){  
 //acessando os erros  
 $http(data.rejection.config).then(**function**(response){  
 //reenviando a requisao  
 **return** data.deferred.resolve(response);  
 });  
 });  
}