

Search plugin for domain class properties. Lightweight plugin which puts the ability for searching, it adds utility functions for building the search result into a string format representation sufficient for auto-complete as well as functions for listing the results based on the search query.

# Quick Search Plugin - Reference Documentation

**Authors:** Matouš Kuera

**Version:** 0.1

## Table of Contents

- 1** Introduction
  - 1.1** Why New Search Plugin
  - 1.2** Change Log
  - 1.3** Roadmap
- 2** Quick Search Service
- 3** Configuration

# 1 Introduction

The Quick Search Plugin is the search implementation suitable for querying one domain class in one query. It supports search of domain class persistent properties as well as embedded and association properties. It is best suited for backend search where the user would like to search in a certain list of domain class objects without specifying which properties should be used for which kind of search. The complex search could be accomplished using [Filter Pane Plugin](#).

## 1.1 Why New Search Plugin

There is certain amount of search and filtering plugins out there, so why we've implemented new plugin. I would like to list the reasons which we understand as the most important advantages of this plugin. But keep in mind, that different plugins are usually suitable for different situations.

- **Search setting should not be specified in domain classes** - In my opinion, specifying the search options (properties which should be searched, the ordering parameters, etc.) should not be specified in domain classes. The domain class represents the model and the search is more view related, therefore it should be specified closer to the view level, i.e. in the controller.
- **Different search settings for different situations** - In addition to above, the search could vary for same domain class list in different situations. The search engine should allow to set different search options though.
- **Let the programmer allow to influence the search** - Sometimes the search should be constrained by programmer specific code. Therefore, we allow to put a custom [create\\_criteria](#) closure to the search query.

## 1.2 Change Log

Date	Version	Notes
2013-11-18	0.1	Initial release. Search service for listing the result domain class objects.

## 1.3 Roadmap

- Tag library for search form with support of autocomplete.

## 2 Quick Search Service

The Quick Search Service is responsible for the whole logic of search. It allows the search based on domain class persistent properties, embedded class properties as well as association properties. It provides two basic methods: `ref/Services/searchServiceSearch` for listing the results and `ref/Services/searchServiceSearchAutoComplete` suitable for listing the results for autocomplete.

### 3 Configuration

There are few configuration options for the plugin:

Name	Default	Description
<code>grails.plugins.quickSearch.search.tokens</code>	<code>''</code>	The strings of characters which should be used for "tokenizing" (in other words splitting) the query string. The default tokenizer is a gap which means if the user search for 'Grails in Action' it will search for 'Grails' OR 'in' OR 'Action'.
<code>grailsApplication.config.grails.plugins.quickSearch.search.tokenizeNumbers</code>	<code>true</code>	The Quick Search plugin supports the search for numbers and strings, this configuration option sets the default behavior of splitting the search queries into numbers and strings. If the user will search for 'Lindwurmstr. 76' the result search would be 'Lindwurmstr.' OR '76'.