

Распознавание типа запроса.

Любой запрос может принадлежать одному из следующих типов:

- Основные проекты - запросы, адресуемые portalу mnogonado.net.
- Поисковые системы - запросы, адресуемые yandex.
- Философский запрос — все остальные.

Запрос может принадлежать одновременно первым двум типам. В этом случае действует правило приоритетов («основные проекты» обладают более высоким приоритетом).

Для распознавания типа используются две структуры:

- дерево структуры запроса
- правила определения типа

Сама процедура распознавания типа двухшаговая:

1. разбор запроса с использованием дерева структуры.
2. Определение типа на основе правил определения типа.

Дерево запроса.

На скриншоте показано дерево запроса. В БД дерево представлено двумя таблицами:

Запрос: SELECT * FROM node_types - типы узлов

Результат-записей: 9

typeid	type_name	comment
1	root	корень дерева запроса
2	question	раздел запроса - ВОПРОС
3	subject	раздел запроса - СУБЪЕКТ
4	action	раздел запроса - ДЕЙСТВИЕ
5	object	раздел запроса - ОБЪЕКТ
6	sub_object	подраздел раздела ОБЪЕКТ
7	concept	понятие
8	synonym	слово, определяющее понятие
9	function	функция, определяющая понятие

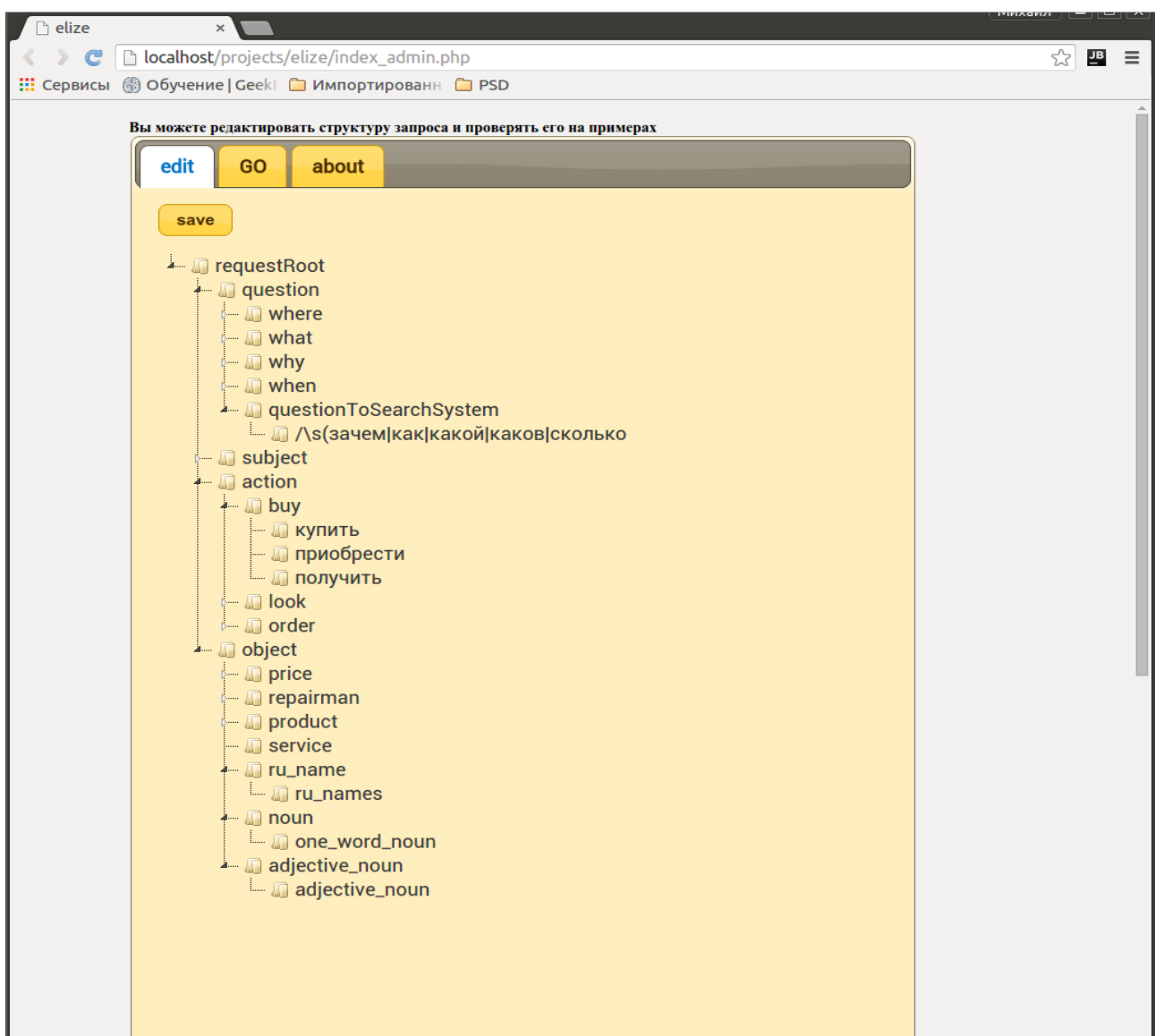
Запрос: SELECT * FROM request_tree - дерево запроса

Результат-записей: 65

nodeid	parentid	node_name	typeid	node_valid	node_default	comment
1		requestRoot	1			КОРЕНЬ запроса
2	1	question	2	1		ВОПРОС запроса
3	1	subject	3	1		СУБЪЕКТ запроса
4	1	action	4	1		ДЕЙСТВИЕ запроса
5	1	object	5	1		ОБЪЕКТ запроса
6	2	where	7	1		вопрос ГДЕ
7	2	what	7	1		вопрос ЧТО
8	2	why	7	1		вопрос ПОЧЕМУ
9	2	when	7	1		вопрос КОГДА
13	3	user	7	1		ПОЛЬЗОВАТЕЛЬ - субъект запроса
14	13	хочу	8	1		синоним USER
15	13	хотелось бы	8	1		синоним USER
16	13	покажите	8	1		синоним USER
17	6	где	8	1		синоним WHERE

18	6	куда	8	1	синоним WHERE
19	6	откуда	8	1	синоним WHERE
20	7	что	8	1	синоним WHAT
21	7	какой	8	1	синоним WHAT
22	7	каков	8	1	синоним WHAT
23	7	который	8	1	синоним WHAT
24	8	почему	8	1	синоним WHY
25	8	зачем	8	1	синоним WHY
26	8	как	8	1	синоним WHY
27	9	когда	8	1	синоним WHEN
28	9	укажите дату	8	1	синоним WHEN
29	4	buy	7	1	действие КУПИТЬ
30	4	look	7	1	действие ПРОСМОТРЕТЬ
31	4	order	7	1	действие ЗАКАЗАТЬ
32	29	купить	8	1	синоним BUY
33	29	приобрести	8	1	синоним BUY
34	29	получить	8	1	синоним BUY
35	30	узнать	8	1	синоним LOOK
36	30	посмотреть	8	1	синоним LOOK
37	30	покажите	8	1	синоним LOOK
38	31	заказать	8	1	синоним ORDER
39	31	заказывать	8	1	синоним ORDER
40	31	сделать заказ	8	1	синоним ORDER
50	5	price	7	1	ЦЕНА__
51	5	repairman	7	1	МАСТЕР
52	5	product	6	1	под-объект ПРОДУКТ, ИЗДЕЛИЕ
53	5	service	6	1	под-объект УСЛУГИ
67	50	цена	8	1	синоним PRICE
68	50	цены	8	1	синоним PRICE
69	50	ценами	8	1	синоним PRICE
70	50	по цене	8	1	синоним PRICE
71	50	по какой цене	8	1	синоним PRICE
72	50	сколько стоит	8	1	синоним PRICE
73	50	стоимость	8	1	синоним PRICE
74	51	мастер по ремонту	8	1	синоним REPAIRMAN
75	51	специалист по ремонту	8	1	синоним REPAIRMAN
76	51	электрик	8	1	синоним REPAIRMAN
77	51	сантехник	8	1	синоним REPAIRMAN
86	52	fridge	7		ХОЛОДИЛЬНИК
87	86	холодильник	8	1	синоним FRIDGE
88	52	washer	7	1	СТИРАЛЬНАЯ МАШИНА
89	88	стиральная машина	8	1	синоним WASHER
91	88	стиральную машину	8	1	синоним WASHER
92	2	questionToSearchSystem	7	1	вопросы к ПОИСКОВОЙ СИСТЕМЕ
94	92	/s(зачем как какой каков сколько где когда который кто куда откуда почему чей что)\s/	8	1	регулярное выражение для запроса к ПС
95	5	ru_name	7	1	имя собственное
96	95	ru_names	9	1	функция, определяющая

					имя собственное
97	5	noun	7	1	существительное
98	97	one_word_noun	9	1	фраза из одного слова - существительного
99	5	adjective_noun	7	1	2 слова: существительное и прилагательное
100	99	adjective_noun	9	1	существительное + прилагательное



Структура дерева описывается следующим образом.

Верхний уровень узлов — это разделы. Всего рассматривается четыре раздела:

- вопрос (что, зачем, как,)
- субъект . В нашей модели не используется

- действие (купить, посмотреть, приобрести,)
- объект («холодильник», «город Москва»,.....)

Для распознавания важны два нижних уровня дерева:
понятие и определение понятия.

Результат разбора — это список понятий, которые удалось определить из текста запроса.

Примеры понятий:

- question/questionToSearchSystem — вопросы к поисковой системе
- action/buy - действие: покупка
- object/ru_name - объект: имя собственное
- object/noun — объект: существительное
- object/adjective_noun — объект: пара существительное и прилагательное

На самом нижнем уровне дерева располагаются определения понятий. Определение может быть двух типов: синоним или функция.

Синоним — это слово или регулярное выражение, определяющее понятие. Если определение понятия задано синонимом, это означает, что в запросе будет искаться слово или какая-то комбинация, удовлетворяющая регулярному выражению.

Функция. Если условие задано именем функции, то при разборе будет обращение к этой функции. Если она возвращает true, то соответствующее понятие попадёт в результат разбора. Пример определения-функции. «ru_name» проверяет является ли слово именем собственным. Соответствующая функция обращается к БД и возвращает true, если слово найдено в таблице имён собственных.

Из сказанного следует, что результат разбора — это список понятий, распознанных на основании дерева структуры запроса. Т.е. что-то следующего вида:
[понятие1, понятие2,].

Например запрос : «купить холодильник в Челябинске» даст следующий результат разбора:
[action/buy(купить),
object/noun(холодильник),
object/ru_name(Челябинске)
]

правила определения типа.

Для каждого типа запроса записывается список понятий. Выглядит это следующим образом.

```
Основные_проекты => [ action/buy,           // купить
                      object/price,         // цена
                      object/noun,          // запрос из одного существительного
                      object/adjective_noun] // запрос из пары слов существительное,
                                           прилагательное
```

```
Поисковые_системы => [question/questionToSerchSystem, //вопросы что, где,когда, какой,....
                      object/ru_name]                 // имя собственное
```

```
Философские_вопросы => []           // всё остальное
```

Алгоритм определения типа. По каждому типу запроса проверяется есть ли в результате разбора понятия, соответствующие данному типу запроса. Если есть хотя бы одно, то считается тип запроса определён.

Скрипты подсистемы распознавания типа запроса.

RequestGo.php - разбор запроса

RequestType.php - определить тип запроса

RequestTree.php - дерево запроса

RequestTree_db.php - взаимодействие с БД

ConceptFunction - функции для определения единиц запроса

MorphologyRu.php - морфологический разбор на основе окончаний слов