# **0:公司简介**

Quixey Technical Overview by Liron <https://www.youtube.com/watch?v=d_3KbV8xCIM>

要点：

Quixey introduces by Tomer Kagan <https://www.youtube.com/watch?v=eEmKQkRtM_o>

要点

ML introduction: <https://www.youtube.com/watch?v=uWWuWLlKwvU>

要点

## **竞争对手**

Baidu shouji

，比如AppFlow, Kinetik, Crosswalk, Discovr Apps,AppsFire, Xyo, Appolicious 和Hubbl 等

functional searching for apps

(smart intelligent stimulating)

do not need to download the apps. Just search and tap and you can use the apps.

only on Android? Apple do not authorize others to search and use apps on IOS. apple is a Closed ecological environment. What is your plan to solve such issue.

2013, sport Android、iOS、BlackBerry 和 Windows Phone 在内的移动平台，包括 Mac os app, Windows app的桌面平台，包括 Chrome、火狐、IE 和 Safari 在内的浏览器平台，以及网页应用。Web、Windows、Mac、iPhone、Android、Firefox、 IE和Chrome

## **职位 Engineer- Knowledge Operations**

o 使用quixey已有平台对海量数据进行分析与处理

o 发现并分析数据异常情况，生成数据报告

o 使用Hadoop/Spark工具对数据质量进行分析与评估

Quixey Beijing－ 爬虫开发实习生

工作职责:

o 负责设计和开发高效的定向爬虫

o 负责网页指定内容的信息抽取、数据清洗工作。

o 负责爬虫数据质量检测和监控工作

o 负责有关爬虫系统的其他模块开发工作

As a Knowledge Operations Engineer you will:

* Creatively find sources for new knowledge and on-board those into the Quixey knowledge system.
* Setup crawls to acquire content from web/api’s
* Participate in the design and development of next-generation knowledge processing systems

As a Knowledge Operations Engineer you will have:

* Experience programming in Java
* Knowledge of JavaScript or some other scripting language
* Familiar with internet protocols as HTTP and FTP
* Extra plusses: Spark, Jenkins, web scraping, AWS.

# **1：http协议**

# **2： web crawler**

PageRank <https://zh.wikipedia.org/wiki/PageRank> (algrithum using hyperlink)

爬video：

1：爬电影名和link:(get url, parsing html(BeautifulSoup), get tag or XPath)

BeautifulSoup(soup.findAll)

data-reactid =64

mechanize vs selenium

# **3: Java**

I use to use java to do my homework to when I was in college.

After I graduate, when I work, I use python to do the project.

# **4: Javascript**

I know the basic javascript, like I know the grammar, the function, object, the event the DOM. Although I am not skilled at it. I am a beginner because I am not involved in angularJS, NodeJS, Ajax, jQuery,

I use JS,html,bootstrap to implemented the front end according to UI design.

I have use JS to implement to form set, which allow user to fill out informations, add more form and delete form.

I have use JS to implement Google Chart.

# **5： spark python**

<http://blog.jobbole.com/86232/>

# **6: 问问题**

1: do you recommend this company?

2:which stage the project is right now and when the project plan to delivery.

3: what is the career ladder for this position for generally in Quixey, i mean, I first work as a knowledge operations engineer and if all goes will that is the next, is is senior knowledge operations engineer or I will be transfer to other group to gain more.

4： if possible, you can give me a spark task, on one hand you can make it as a test, on other hand, I can do the practice.

Search Content Pipelines

- Content Pipelines

- Knowledge Piplines

difference and connect with knowledge, content,data

1: how many people in this team

2: what web crawl technology are used in our crawler. do you use scrapy? because scrapy is what I want to involve next.

爱奇艺， 优酷，豆瓣，百度movie。腾讯视频

抓数据时：

1：用什么工具，优缺点，为什么选这个工具，

normalization：

1：follow 的rule是什么，遇到的困难是什么，怎么样解决的， 什么样的用户用

他在这家公司工作是一种什么体验。

接着，询问他目前的工作。

google instant app

# 7 glassdoor

describe the pipeline of building a spam email detection system

*Detect cycle in linked list(The turtle and hen method.)*

*mixed bag comprising of fairly standard*

Explain what challenges you encountered while designing the schema

coding challenge involved data structure problem involving array handling

function to recursively traverse a directory structure to access all files (in all subdirectories)

program to measure angle between hour and minute hand of a clock

program to count occurrences of a characters in an input stream

come up with test cases for a restful api

qa process related questions

Given three points, write code to determine whether they are collinear

1. 给一个数组，找里面identical pairs的数目。举例f([2,0,1,0])=1，(1,3)和(3,1)算一个。

取set,然后=n\*（n-1）/2，然后有几个是重复的数再加几

2. 如果一个数组里*a [ i ]+a[j]=k ，它俩就是k-complementary。找这个数组里k-complementary的个数。举例f(8, [2,3,4,5])=3，(1,3)和(3,1)算两个，(2,2)也算。*

*这题类似于two sum*

*. more info on 1point3acres.com*

*补充内容 (2015-2-25 16:06):*

*过了 说是把简历转给了Hiring Managers*

*补充内容 (2015-3-5 17:24):. 涓€浜�-涓夊垎-鍦帮紝鐙鍙戝竷*

*说是要安排一个Hiring Manager电面45分钟，这个一般是聊啥，简历么？*

*1. k window max in array*

*2. topological sorting*

*3. give a list of intervals, find a point intersect with most ittervals.*

*[0,2],[1.5],[3,6],[4,7],[7,8]*

*让其排列成 [(0,s),(1,s),(2,e),(3,s),(4,s),(5,e),(6,e),(7,s),(7,e),(8,e)]*

*然后count=0，和一个array，array里面是每一个count的数；其中遇到s,count+=1,遇到e,再下一个点-=1*

*于是就有array = [1,2,2,2,3,3,2,2,0]，求max(array)*

*merge sort/ quick sort*

*有一题topological sort*

*给你两个数组， 一个代表一行排队的人的高度*

*一个代表每个人前面比他高的人数*

*让你根据这俩个数组决定次序*

*比如*

*高度 3 1 4 5*

*人数 1 2 1 0*

*那么正确的排队次序是这样的:*

*5 3 4 1*

*0 1 1 2*

*例子里面应该是：*

*: 高度 3 1 4 5*

*: 人数 2 2 1 0*

*吧？*

*是不是打错了？*

*这题不是应该用segment tree解吗？*

*考古一下，板上看见好多回了*

*有点像 russian doll envelops*

[*https://leetcode.com/problems/russian-doll-envelopes/*](https://leetcode.com/problems/russian-doll-envelopes/)

Elasticsearch

deep learning models

Specialties:

. Tackle Hardest problems around

. Mobile Deep Search with actions (Skills), Automated generation of knowledge / Hero like cards from apps and their search functions

- Static and dynamic analysis tools to find all skills / functions/ actions and Deep Links inside apps

- Tools for semi Auto tagging of functions with entities and actions inside apps along with user targeted intents

- End to End Mobile + Server performance

- Mobile Devices Security

- Setting up, managing large global engineering teams. Driving them to ensure delivery of the product. Interaction with Product teams

- Architect, Engineering Leader with proven record for delivering complex, large technical products

- Building and Hosting custom android emulators in the 'cloud' and letting users preview the apps answer their queries without installing and serving app's native content in native look similar to Knowledge / Hero cards (Think Google instant apps + lightweight UIXML for rendering). Building similar tools and infrastructure for iOS Simulators

- Static and dynamic analysis tools to explore Deep Linking inside apps

- Tools for deep app content acquisition from Apps directly

- Android Security and Isolation techniques (app virtualization via binary rewriting , seandroid , trust zone)

. Mobile performance

- Chief SW Architect and Engineering Lead for device engineering for Knox. (www.samsung.com/knox)

- Developed the android isolation technique based on app wrapping but backed with framework based security isolation guarantee based on namespaces and SEAndroid

- In Depth knowledge of Mobile Operating systems, low level optimizations

- Multicore Expertise, Scheduling algorithms and Parallelisation of Complex systems

- Operating system with special focus for parallel and multicore devices.

- Automatic scheduling algorithms for both static and dynamic solutions and automated profilers. Implemented a real time operating system for multicore.

Tools for DeepSearch :

• Mobile Deep Search, Automated generation of knowledge / Hero like cards from apps and their search functions

• Semi Auto tagging of functions with entities and actions inside apps along with user targeted intent tagging

• Building static and dynamic analysis tools to explore deep linking inside apps (both Android and iOS) and identify search functions, entity functions and various actions(skills) in apps

• Building analysis tools to identify useful actions and functions inside apps (both Android and iOS)

• Building and Hosting custom android emulators in the 'cloud' and letting users preview the apps answer their queries without installing and serving app's native content in native look similar to Knowledge / Hero cards (Think instant apps + lightweight UIXML for rendering). Building similar tools and infrastructure for iOS Simulators

• Building innovative tools to acquire deep app content via crawling native apps and real time fetching (both Android and iOS)

• Content Processing, Structured content from Unstructured / raw data and Knowledge Systems

• Exploring presentation solutions for mobile / deep search

--------------------------------------------------------

Leading Core Infrastructure at Quixey - Not only devops, this means distributed serving systems, data/content/knowledge platform and the CI pipeline:

Building software, leading change.

All software deployed in public clouds: AWS and Aliyun (China)

Responsible for building:

. Data Engineering Platform - clickstream, incremental data aggregation, presentation (Kafka, Spark. SparkSQL),

. Content Processing Platform (Spark based dataflow engine),

. Customer end-point Infrastructure

. Services Orchestration (Apache Camel), Discovery (Consul, Zookeeper)

. Services Container

. Dev Platform ('devops', Continuous Integration, Docker, Container Deployment / Orchestration, ...)

. Test Platform - test automation in our CI pipeline

. Knowledge Engineering (Spark)

• Implemented Chinese query tokenizer using NLP knowledge.

• Applied Early parser to avoid backtracking.

• Applied dynamic programming in chart parsing to reduce processing time to linear.

• 40% faster and 1.6% gain of DCG score compared with current tokenizer.

• Has been integrated to Quixey app search engine.

• Intellectual property application has been accepted and in process.

Search Relevance

- Ranking features for App/Mobile Search

- Ranking Algorithms

- Search Quality Metrics

Search Engine Backend

- Query Understanding

- ElasticSearch Cluster

Search Content Pipelines

- Content Pipelines

- Knowledge Piplines

Engineering Director for offline content processing and Knowledge Base. I lead the implementation and operation of Quixey's Spark based content processing pipelines as well as the Knowledge Base. I have build a high performance engineering team who very successfully have build and operated the first two generations of the systems.

被提问：

1:refresh the browser, what happen：

Functionally what happens is

the browser will ask the server for a fresh version of the document located at the address described in the address bar of the browser. The request for that document is much more than just a URL.

It also has headers that the browser appends that will tell the server who you are, what kind of device you are using, the type and version of the browser, installed plugins, cookies that match the domain and more.

All of this information (which can sometimes be a LOT of information, like MBs of data) is send in the request.

The next thing is that request has to find the server at the specified address.

Since you have been to this page before, and assuming default settings, the domain will be looked up in the DNS cache already stored on your computer.

Now the request is broken into packets and the first packet goes to the router that assigned you your local ip.

If your on wifi that means your wireless modem starts to interact with the tcp/ip software driver installed on you operating system to turn you message into a (hopefully encrypted) wireless signal that propagates via a mechanism I don't fully understand through space at the speed of light in every direction.

If you are within range that signal is detected by the router that signals back that it got the packet and is ready for the next.

Back and forth they talk till the request is sent up the chain from your home router, to the local neighborhood router your ISP set up, to the area router, to the ISPs main connection to the Internet backbone then to top level routers then based on the ip range back down through a series of routers to the server.

The server signals it got the first packet than asks for the next one.

Back and forth they talk till the entire request is uploaded to the server.

The server than has a lot of work to do. It reads the packet, and a web server processes a lot of rules to figure how to respond.

Sometimes it looks at the time stamp of the request, compared it to the document you already have and say "nothing changed" that message goes back through the Internet to your browser that will then just read its local cached version of the resonse and all the files referenced in that response.

Sometimes it routes the request to another server or software on the server like a content management system. It will read the headers and process its own rules. It too says nothing has changed. The browser reads the cache and redraws the screen. I'm not going to go any further because it's dinner time.

**TCP/IP stack**

* Application layer creates a HTTP request, packs it into a TCP packet, and sends it to the Transport layer.
* Transport layer packs the TCP packet into an IP packet and sends it to the Network layer.
* Network layer forwards the packet based on its destination IP and sends the packet along with the address of the next network node to the Data-Link layer.
* Data-Link layer transfers the packet to the next node via the Physical layer.
* Physical layer propagates the bits of the packet along a wire back to the Data-Link layer

Then the packet will crawl its way back the TCP/IP stack to the web server. Then a new copy of the page will crawl its way back down and up the TCP/IP stack back to your Web Browser.

if the request method is GET: 无害

if the request method is POST: 数据会被重新提交

get post

cache

which server

token:systimestamp+ID