PYCON 2019 IN THAILAND

SPEAKER: SEO, JONGHWA

DJANGO FOR E-COMMERCE

INTRODUCTION TO MYSELF

- Thailand
  - taught in Naresuan University for 2 years
  - worked in Bangkok for 2 years
- Korea
  - running my own business
    - pincoin.co.kr (based on Django / Python)
      - # of Users: 35,000+
      - Monthly Transactions: 13,000+
      - Monthly PV: 344,000+
      - Concurrent Visits: ~70
    - smartstore.naver.com/pincoin (hosted by Naver)
    - github.com/pincoin/rakmai

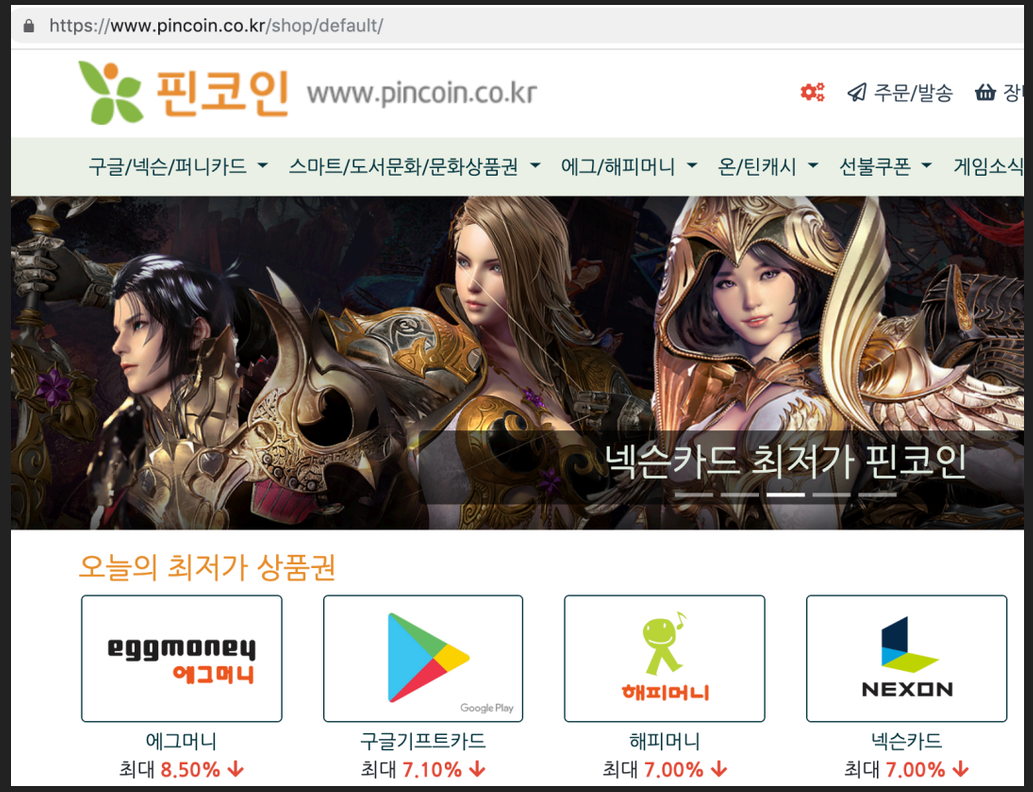

TABLE OF CONTENTS

DJANGO (1/2)

- The web framework for perfectionists with deadlines
  - web framework
  - deadlines
- Nowadays, we have to learn framework with language.
  - Django, Flask / Python
  - Spring / Java
  - Wordpress, Laravel, Codeigniter / PHP
  - Rails / Ruby
- Questions
  - What do Korea or Thailand use the most?
  - Which one will you choose?

DJANGO (2/2)

- Django concerns
  - Ridiculous fast
    - It helps develop fast, not performance.
- Reassuring secure
  - It helps avoid common security mistakes.
- Exceedingly scalable
  - It helps scale out.

E-COMMERCE (1/3)

- Mall types
  - Open market stores
    - smartstore.naver.com/pincoin
    - shoppee.co.th/store
- Your domain: subscribe service plan
  - cafe24.com, makeshop.co.kr, godo.co.kr
- Your domain: build your own store using OSS or else
  - wordpress, opencart, prestashop, magenta
  - Django / Python!

E-COMMERCE (2/3)

- Cart Features
  - Product/Stock management
    - Options, attributes and complex relationships
    - Review and ratings
  - Customer/Membership management
    - Discount, special offer,
    - Reward points
- Cart
- Payment
- Shipping
- Admin reports

Maintenance and Security

E-COMMERCE (3/3)

- Why Django?
  - Django / Python enforces coding conventions / styles.
    - There should be one – and preferably only one – obvious way to do it
    - Maintenance, Maintenance, Maintenance!
  - ORM
  - Admin page off-the-shelf
  - Security support: https://docs.djangoproject.com/en/2.1/topics/security/
  - Lots of 3rd party apps support
    - django-allauth
    - django-mptt and django-model-utils
    - djangorestframework
- Rule of Thumb: I was able to build my website by myself thanks to Django.

DJANGO TECH STACK FOR E-COMMERCE (1/4)

- Django does NOT do EVERYTHING for service.
  - PIP packages
  - OS, Daemons and Frontend
  - Other Services

DJANGO TECH STACK FOR E-COMMERCE (2/4)

- PIP Packages(requirements.txt)
  - django-allauth
  - djangorestframework
  - django-model-utils
  - django-mptt
  - django-otp
  - django-crispy-forms
  - django-ipware
  - geoip2
  - django-taggit
  - django-import-export

DJANGO TECH STACK FOR E-COMMERCE (3/4)

- OS, Daemons and Frontend
    - Ubuntu (AWS Lightsail)
    - NGINX, uWSGI
    - PostgreSQL
    - Memcached
    - Celery, RabbitMQ
    - crontab
    - jQuery 3, Bootstrap 4, FontAwesome 5
    - Android app to forward SMS payment notification

DJANGO TECH STACK FOR E-COMMERCE (4/4)

- Other Services
  - Mailgun
  - Line notify
  - Cloudflare WAF
  - disq.us
  - Google Analytics
  - Aligo SMS (KR)

BACKGROUND FOR DJANGO / PYTHON (1/2)

- Object Oriented Programming
  - Data encapsulation and Data abstraction
  - Inheritance and Polymorphism
    - View Mixins, Templates, Model Inheritance
- ORM(Object Relation Mapping) - Save from SQL Injection attacks
  - Table = Class
  - Record(Row) = Instance(object)
  - Column(Field) = Member variable

BACKGROUND FOR DJANGO / PYTHON (2/2)

- Design Patterns https://github.com/faif/python-patterns
  - MVC (model, view, controller) - Model Template View
  - Decorator (is-a, inheritance)
  - Composite (has-a, composition)
  - Observer (signals)
  - Adapter
  - Factory
  - Singleton
- Test Driven Development www.obeythetestinggoat.com

DJANGO PRACTICES FOR E-COMMERCE (1/10)

- Model
  - Extending abstract model
    - https://github.com/pincoin/rakmai/blob/master/rakmai/models.py#L100-L101
  - django-mptt : MPTTModel
    - https://github.com/pincoin/rakmai/blob/master/rakmai/models.py#L104-L132
  - django-model-utils: Choices, TimestampedModel, SoftDeletableModel
    - https://github.com/pincoin/rakmai/blob/master/shop/models.py#L25-L29
- Relationships
  - one-to-one
    - https://github.com/pincoin/rakmai/blob/master/member/models.py#L38-L41
  - many-to-one (foreign key)
    - https://github.com/pincoin/rakmai/blob/master/shop/models.py#L100-L106
  - many-to-many
    - https://github.com/pincoin/rakmai/blob/master/shop/models.py#L334

DJANGO PRACTICES FOR E-COMMERCE (2/10)

- Forms
  - Validation (super duper important!)
    - Cart!
    - https://github.com/pincoin/rakmai/blob/master/blog/forms.py#L122-L130
  - Model form
    - https://github.com/pincoin/rakmai/blob/master/help/forms.py#L101-L115
- Model form for Admin
  - https://github.com/pincoin/rakmai/blob/master/shop/forms.py#L7-L28
- Passing arguments to Form
  - https://github.com/pincoin/rakmai/blob/master/shop/views.py#L229-L234
- django-crispy-forms
  - It helps create a simple form easily, but sometimes makes more difficult
    - https://github.com/pincoin/rakmai/blob/master/blog/forms.py#L45-L77
  - If you've got lots of JS codes, it could be better to render forms manually.

DJANGO PRACTICES FOR E-COMMERCE (3/10)

- Widgets
  - editors / form assets
    - summernote https://github.com/pincoin/rakmai/blob/master/rakmai/widgets.py#L57-L71
  - render html
    - https://github.com/pincoin/rakmai/blob/master/rakmai/templates/summernote/summernote.html
  -

DJANGO PRACTICES FOR E-COMMERCE (4/10)

- Templates
  - ‘extends‘
  - custom template filter
    - currency https://github.com/pincoin/rakmai/blob/master/shop/templatetags/shop_filters.py#L10-L16
- custom template tags
  - simple tag https://github.com/pincoin/rakmai/blob/master/shop/templatetags/shop_tags.py#L82-L99
  - recursive tag https://github.com/pincoin/rakmai/blob/master/shop/templatetags/shop_tags.py#L17-L57
-

DJANGO PRACTICES FOR E-COMMERCE (5/10)

- View
  - Class-based View (CBV) vs. Function-based View(FBV)
    - I don’t use FBV.
  - MRO (Method Resolution Order)
    - List, HostRestricted, LoginRequired, Pageable https://github.com/pincoin/rakmai/blob/master/shop/views.py#L409
    - FormView https://github.com/pincoin/rakmai/blob/master/shop/views.py#L556-L592
- ViewMixins
  - ViewMixins https://github.com/pincoin/rakmai/blob/master/shop/viewmixins.py#L11-L48

DJANGO PRACTICES FOR E-COMMERCE (6/10)

- REST API
  - JSON response
    - JsonResponse https://github.com/pincoin/rakmai/blob/master/shop/views.py#L376-L381
  - APIView
    - Response https://github.com/pincoin/rakmai/blob/master/member/views.py#L526-L533
    - Serializer https://github.com/pincoin/rakmai/blob/master/api/serializers.py
    - Permission https://github.com/pincoin/rakmai/blob/master/api/permissions.py

DJANGO PRACTICES FOR E-COMMERCE (7/10)

- Security
  - SSL/TLS
    - https://docs.djangoproject.com/en/2.2/topics/security/#ssl-https
    - Use Cloudflare
- Session
  - cart https://github.com/pincoin/rakmai/blob/master/shop/helpers.py#L32-L136
  - cart-add https://github.com/pincoin/rakmai/blob/master/shop/views.py#L326-L329

DJANGO PRACTICES FOR E-COMMERCE (8/10)

- Caching
  - https://github.com/pincoin/rakmai/blob/master/shop/templatetags/shop_tags.py#L62-L79
- Middleware
  - https://github.com/pincoin/rakmai/blob/master/rakmai/middleware.py
  - https://github.com/pincoin/rakmai/blob/master/shop/middleware.py

DJANGO PRACTICES FOR E-COMMERCE (9/10)

- Celery
  - RabbitMQ
  - Celery workers
    - Register celery app
      - https://github.com/pincoin/rakmai/blob/master/sandbox/__init__.py
    - Celery tasks
      - https://github.com/pincoin/rakmai/blob/master/sandbox/celery.py
    - Tasks
      - https://github.com/pincoin/rakmai/blob/master/shop/tasks.py

DJANGO PRACTICES FOR E-COMMERCE (10/10)

- Admin Page
  - https://github.com/pincoin/rakmai/blob/master/shop/admin.py

- How develop
  - Developers work with customers using models and admin pages.
  - Designers work with customers using templates.
  - Finally, developers and designers collaborate by working with views.

OTHER SERVICES (1/3)

- OS
  - VPS
    - AWS Lightsail
    - VULTR
    - Digital Ocean
- Cloud
  - AWS
  - Azure

OTHER SERVICES (2/3)

- Cloudflare
  - DNS (Hide your server IP)
  - SSL/TLS
  - CDN, minify, caching
  - Rewrite Rule
  - Web Application Firewall

OTHER SERVICES (3/3)

- Mail Service
  - Mailgun, G-Suite
- Legacy
  - Postfix
  - Dovecot
  - Roundcube

CONCLUSION (1/3)

- Django can be used to develop any web services.
  - Rapid development
  - Security
  - Scalability
  - Software maintainability

CONCLUSION (2/3)

- References
  - Django office document

    docs.djangoproject.com/en/2.2/
  - Two scoops of Django 1.11

    www.twoscoopspress.com
  - Test-Driven Development with Python, 2nd

    www.obeythetestinggoat.com
  - pincoin

    github.com/pincoin/rakmai

CONCLUSION (3/3)

- Any Questions?
- Thank you.