

# The School of the Islamic Republic of Iran

100 Carlton Vale, London NW6 5HE

**Inspection date**

1 March 2022

**Overall outcome**

**The school does not meet all of the independent school standards that were checked during this inspection**

## Main inspection findings

### Part 1. Quality of education provided

*Paragraphs 2(1), 2(1)(a), 2(1)(b), 2(1)(b)(i)*

- At the time of the last inspection leaders had bought new schemes of work. However, these had been taken from published sources and not adapted or implemented effectively. Teachers also did not deliver the work that was planned.
- There are now schemes of work for several subjects. Some of these are from published sources and they are detailed. However, schemes of work are not in place for all subjects and for all year groups. In some subjects, for example computing and physical education (PE), there are no plans at all. It is not clear what will be taught in these lessons and what leaders want pupils to know and remember.
- Where schemes of work exist, they consider the ages and needs of all pupils. However, this is not the case in subjects where there are no schemes of work. For these subjects, leaders have only drawn up a statement of the national curriculum aims. Leaders have not thought about what knowledge they want pupils to know and how this will be taught. It is, therefore, also unclear whether the curriculum will take into account the needs of all pupils.
- The standards in this part continue to not be met. The school is repeatedly failing to meet the same standards over time.

*Paragraphs 2(2), 2(2)(d), 2(2)(d)(i), 2(2)(d)(ii), 2(2)(e), 2(2)(e)(i), 2(2)(e)(ii), 2(2)(e)(iii), 2(2)(h) and 2(2)(i)*

- At the time of the previous inspection, too little time and guidance were given to personal, social, health and economic (PSHE) education. Specific characteristics, such as sexual orientation, were not included in the school's equalities policy or considered within PSHE. Leaders were also unable to provide details about the school's careers programme.
- Leaders now have a PSHE programme that reflects the school's aim and ethos and encourages respect for others. Leaders encourage respect for other people, paying particular regard to the protected characteristics. They also actively promote mutual

respect and tolerance of those with different faiths and beliefs. Pupils are encouraged to respect other people equally.

- Leaders have now created a careers plan that provides pupils with independent careers education, advice and guidance. This programme provides information about a wide range of potential careers. Pupils receive sufficient information to prepare them for the opportunities and responsibilities of life in modern Britain.
- The standards in this part are now met.

*Paragraph 3, 3(a), 3(b), 3(c), 3(d), 3(f) and 3(g)*

- During the previous inspection, inspectors noted that teachers were not given enough guidance or professional development. They were not taught how to adapt activities to ensure that pupils learn well, and new teachers did not receive adequate training about how to teach effectively.
- Inspectors visited lessons in a range of subjects and spoke to pupils and teachers. Teachers have the resources they need to plan and deliver lessons that enable pupils to progress through the curriculum that is in place. Pupils are keen to learn and they are motivated in class. Teachers assess what pupils know and remember and they address any gaps in pupils' understanding.
- The standards in this part are now met.

## Part 2. Spiritual, moral, social and cultural development of pupils

*Paragraph 5, 5(b) and 5(b)(vi)*

- At the time of the previous inspection leaders were not giving sufficient time to the teaching of PSHE. Specific characteristics, such as sexual orientation, were not included in the school's policy or considered within PSHE.
- The PSHE programme has been rewritten. Pupils have been taught about different types of family and relationships. They are encouraged to respect other people and other people's choices, including when they are different to their own.
- The standards in this part are now met.

## Part 3. Welfare, health and safety of pupils

*Paragraph 7, 7(a), 7(b)*

- Arrangements for safeguarding were judged ineffective at the last inspection. Although most staff were aware of different forms of abuse and knew how to refer concerns, new staff had not received training.
- Safeguarding remains ineffective at the time of this inspection. There are two safeguarding leaders and they have not updated their training since 2019 and it has now expired. Staff now receive termly safeguarding training. They know how to identify signs of abuse and they understand their responsibility to keep pupils safe. However, the training they have received from leaders has not included peer-on-peer abuse. This is a statutory requirement that is not currently being met.
- The standards in this part continue to not be met. The school is repeatedly failing to meet the same standards over time.

### *Paragraph 15*

- At the time of the previous inspection, leaders showed different versions of the school admissions register to inspectors. Leaders could not confirm whether information about pupils taken off roll had been properly recorded and that the relevant authorities had been informed.
- Leaders have addressed these issues and the admission and attendance register is now maintained in accordance with the Education (Pupil Registration) (England) Regulations 2006.
- The standard in this part is now met.

### *Paragraph 16, 16(b)*

- Leaders have a risk assessment policy and they complete risk assessments for several different activities, for example, when pupils go on school trips. These risk assessments identify the different types of risk that pupils might encounter. However, staff completing these forms do not identify any strategies to mitigate these risks. The risk assessment form does not provide staff with an opportunity to include the mitigating actions they intend to take.
- The standard in this part is now not met.

## Part 4. Suitability of staff, supply staff, and proprietors

### *Paragraph 18(2), 18(2)(b), 20(6), 21(3), 21(3)(a), 21(3)(a)(iii), 21(3)(b)*

- At the previous inspection, leaders had not ensured that all necessary checks were carried out on staff when they were appointed to the school. This was also the case for members of the governing body, except the chair.
- Leaders now maintain records of all checks that they complete on staff, governors and the proprietor. Leaders do not employ supply staff. However, leaders have still not completed all the checks required on the staff they employ. They have not completed prohibition order or section 128 checks for staff or the proprietor.
- The standards in these parts are still unmet. The school is repeatedly failing to meet the same standards over time.

### *Paragraph 18(2)(a), 18(2)(c), 18(2)(c)(i), 18(2)(c)(iii), 18(2)(c)(iv), 18(2)(d), 18(2)(e), 21(3)(a)(i), 21(3)(a)(ii), 21(3)(a)(iv), 21(3)(a)(v), 21(3)(a)(vi), 21(3)(a)(vii), 21(3)(a)(viii), 21(4), 21(6), 21(7), 21(7)(a) and 21(7)(b).*

- At the time of the previous inspection, leaders were not completing many of the required checks on staff. For example, they were not checking the identity of staff, the medical fitness of staff or their right to work in the United Kingdom. Leaders now complete more staff checks than at the time of the previous inspection. Therefore, more of the standards in this part are met compared to previously.
- The standards in these parts are now met.

## Part 5. Premises of and accommodation at schools

### *Paragraph 23(1), 23(1)(c),*

- At the time of the previous inspection changing facilities for PE were not suitable. Only one shower cubicle was available for pupils' use after PE. This was in a staff toilet.

- The location of the changing facility and shower is no longer a staff toilet. However, this changing facility cannot be locked for privacy from the inside and can only be locked by key from the outside. This is not suitable.
- The standards in this part are still not met.

*Paragraph 24(1), 24(1)(b), 25*

- There is a medical room on the school's premises. However, this medical room does not contain a sink. The nearest washing facility is next door in the changing room. However, this room is routinely kept locked or could be in use by pupils changing. Therefore, the medical room does not meet requirements.
- Leaders do not ensure that the school's premises and accommodation facilities are maintained to a standard that ensures the health, safety and welfare of pupils. For example, recent fire-safety checks had not identified that one of the fire extinguishers could not be accessed. The school has closed-circuit television cameras, but governors have not officially approved the policy for this. Leaders did not give inspectors any documentation for the control of hazardous substances used in the school's science laboratory.
- The standard in this part is now not met.

*Paragraph 27, 27(b), 29(1), 29(1)(a)*

- At the time of the previous inspection, there was not adequate external lighting to enable people to enter and leave the premises safely. Furthermore, there was no suitable outside space to enable pupils to do PE in accordance with the curriculum.
- Leaders have installed external lighting that allows for the safe entry and exit to and from the premises and there is also suitable outside space for pupils to do PE.
- The standards in these parts are now met.

## Part 6. Provision of information

*Paragraph 32(1), 32(1)(d), 32(2), 32(2)(b), 32(2)(b)(i), 32(2)(c), 32(3)(g)*

- The standards in this part were met at the previous inspection. However, there is some information that is not available now.
- Leaders maintain a school website that contains important information for parents and carers. However, previous inspection reports are not available on the website. The names of all members of the governing body are on the website. However, it is not clear who is the chair of the governing body, and there are no contact details for him. In addition, there are no contact details for the headteacher outside of term time.
- The standards in this part are now not met.

## Part 7. Manner in which complaints are handled

*All paragraphs*

- There is a complaints policy on the school website. This policy allows for complaints to be handled informally. Where a parent is not happy with the outcome of an informal process, there is a formal procedure for parents to follow. There are clear timescales for all types of complaints and there are systems to keep records of all complaints received.
- The standards in this part are met.

## Part 8. Quality of leadership in and management of schools

### *Paragraph 34(1), 34(1)(a) and 34(1)(b)*

- At the time of the previous inspection, the proprietor, leaders and governors had not ensured that all the independent school standards were met.
- Leaders have made some progress in ensuring that some of the standards previously not met are now met. However, there is still a significant number of standards that remain not met. Leaders did not demonstrate that they had the good knowledge and skills necessary to ensure the independent school standards are met consistently.
- The standards in this part are still not met. The school is repeatedly failing to meet the same standards over time.

### Schedule 10 of the Equality Act 2010

- Leaders were unable to provide inspectors with an accessibility plan. Therefore, leaders have not made the arrangements necessary to meet the requirements of schedule 10 of the Equality Act 2010.
- The standard in this part is not met.

## **Compliance with regulatory requirements**

The school does not meet the requirements of the schedule to The Education (Independent School Standards) Regulations 2014 ('the independent school standards') and associated requirements that were checked during this inspection, as set out in the annex of this report. This included the standards and requirements that the school was judged to not comply with at the previous inspection. Not all of the standards and associated requirements were checked during this inspection.

## School details

Unique reference number	133385
DfE registration number	304/6079
Inspection number	10213900

This inspection was carried out under section 109(1) and (2) of the Education and Skills Act 2008, the purpose of which is to advise the Secretary of State for Education about the school's suitability for continued registration as an independent school.

Type of school	Other independent school
School status	Independent school
Age range of pupils	6 to 16
Gender of pupils	Mixed
Gender of pupils in the sixth form	Mixed
Number of pupils on the school roll	120
Number of part-time pupils	None
Proprietor	Seyed Abbas Hosseini
Chair	M Moghadas
Headteacher	Seyed Abbas Hosseini
Annual fees (day pupils)	£850
Telephone number	0207 372 8051
Website	<a href="http://www.iris.school.co.uk">www.iris.school.co.uk</a>
Email address	<a href="mailto:info@irischool.co.uk">info@irischool.co.uk</a>
Date of previous standard inspection	10 to 12 December 2019

## Information about this school

- This is an Islamic independent school. The school is registered for 81 pupils aged six to 16. At present, 120 pupils aged five to 16 are on roll. The school is operating beyond its registration agreement with the Department for Education (DfE) as it is recorded on the Get Information about Schools website, and it has not notified the DfE. It has admitted pupils below the registered age range and has exceeded the maximum number of pupils on roll that it is registered to admit.

- The school was last inspected in December 2019, when it was judged to be inadequate. The inspection judged that the school did not meet the Education (Independent School Standards) Regulations 2014 and associated requirements.
- The school operates from 100 Carlton Vale, London, NW6 5HE
- Leaders produced action plans to address non-compliance with the independent school standards in March 2019 and following the standard inspection in December 2019. Both these action plans were deemed as unacceptable.
- The school does not have any pupils with an education, health and care plan. The school does not identify any pupils as having special educational needs and/or disabilities.
- The school does not use alternative providers.
- The proprietor is also the headteacher. He is supported by a board of governors.



## Information about this inspection

- This inspection was carried out at the request of the registration authority for independent schools. The purpose of the inspection was to monitor the progress the school has made in meeting the independent school standards and other requirements that it was judged to not comply with at its previous inspection.
- The inspection was conducted without notice.
- The DfE requested that this progress monitoring inspection consider all the independent school standards.
- Inspectors met the headteacher, who is also the proprietor, the primary headteacher, teachers and pupils. Inspectors also met with the leader responsible for safeguarding.
- Inspectors toured the premises and reviewed key documents and policies in relation to the independent school standards.
- Inspectors visited lessons and observed pupils during break and lunch.

## Inspection team

Mark Smith, lead inspector

Her Majesty's Inspector

Sophie Healey-Welch

Her Majesty's Inspector

## **Annex. Compliance with regulatory requirements**

### **The school does not meet the following independent school standards**

*Standards that were not met at the previous inspection and remain un-met at this inspection*

#### **Part 1. Quality of education provided**

- 2(1) The standard in this paragraph is met if–
  - 2(1)(a) the proprietor ensures that a written policy on the curriculum, supported by appropriate plans and schemes of work, which provides for the matters specified in sub-paragraph (2) is drawn up and implemented effectively; and
  - 2(1)(b) the written policy, plans and schemes of work–
  - 2(1)(b)(i) take into account the ages, aptitudes and needs of all pupils, including those pupils with an EHC plan.

#### **Part 3. Welfare, health and safety of pupils**

- 7 The standard in this paragraph is met if the proprietor ensures that–
  - 7(a) arrangements are made to safeguard and promote the welfare of pupils at the school; and
  - 7(b) such arrangements have regard to any guidance issued by the Secretary of State.

#### **Part 4. Suitability of staff, supply staff, and proprietors**

- 18(2) The standard in this paragraph is met if–
  - 18(2)(b) no such person carries out work, or intends to carry out work, at the school in contravention of a prohibition order, an interim prohibition order, or any direction made under section 128 of the 2008 Act or section 142 of the 2002 Act, or any disqualification, prohibition or restriction which takes effect as if contained in either such direction.
- 20(6) The standard in this paragraph is met in relation to an individual ("MB"), not being the Chair of the school, who is a member of a body of persons corporate or unincorporate named as the proprietor of the school in the register or in an application to enter the school in the register.
- 21(3) The information referred to in this sub-paragraph is–
  - 21(3)(a) in relation to each member of staff ("S") appointed on or after 1<sup>st</sup> May 2007, whether–
  - 21(3)(a)(iii) a check was made to establish whether S is subject to any direction made under section 128 of the 2008 Act or section 142 of the 2002 Act or any disqualification, prohibition or restriction which takes effect as if contained in such a direction
  - including the date on which each such check was completed or the certificate obtained; and
  - 21(3)(b) in relation to each member of staff ("S"), whether a check was made to establish whether S is subject to a prohibition order or an interim prohibition order, including the date on which such check was completed.

## **Part 5. Premises of and accommodation at schools**

- 23(1) Subject to sub-paragraph (2), the standard in this paragraph is met if the proprietor ensures that–
- 23(1)(c) suitable changing accommodation and showers are provided for pupils aged 11 years or over at the start of the school year who receive physical education.

## **Part 8. Quality of leadership in and management of schools**

- 34(1) The standard about the quality of leadership and management is met if the proprietor ensures that persons with leadership and management responsibilities at the school–
- 34(1)(a) demonstrate good skills and knowledge appropriate to their role so that the independent school standards are met consistently
- 34(1)(b) fulfil their responsibilities effectively so that the independent school standards are met consistently.

*Standards that were met at the previous inspection, but are now judged to not be met at this inspection*

## **Part 1. Quality of education provided**

- 2(2) For the purposes of paragraph (2)(1)(a), the matters are–
- 2(2)(a) full-time supervised education for pupils of compulsory school age (construed in accordance with section 8 of the Education Act 1996), which gives pupils experience in linguistic, mathematical, scientific, technological, human and social, physical and aesthetic and creative education.

## **Part 3. Welfare, health and safety of pupils**

- 16 The standard in this paragraph is met if the proprietor ensures that–
- 16(b) appropriate action is taken to reduce risks that are identified.

## **Part 4. Suitability of staff, supply staff, and proprietors**

- 20(6) The standard in this paragraph is met in relation to an individual ("MB"), not being the Chair of the school, who is a member of a body of persons corporate or unincorporate named as the proprietor of the school in the register or in an application to enter the school in the register, if–
- 20(6)(a) MB–
- 20(6)(a)(ii) does not carry out work, or intend to carry out work, at the school in contravention of a prohibition order, an interim prohibition order, or any direction made under section 128 of the 2008 Act or section 142 of the 2002 Act or any disqualification, prohibition or restriction which takes effect as if contained in either such direction.

## Part 5. Premises of and accommodation at schools

- 24(1) The standard in this paragraph is met if the proprietor ensures that suitable accommodation is provided in order to cater for the medical and therapy needs of pupils, including–
  - 24(1)(b) accommodation for the short-term care of sick and injured pupils, which includes a washing facility and is near to a toilet facility.
- 25 The standard in this paragraph is met if the proprietor ensures that the school premises and the accommodation and facilities provided therein are maintained to a standard such that, so far as is reasonably practicable, the health, safety and welfare of pupils are ensured.

## Part 6. Provision of information

- 32(1) The standard about the provision of information by the school is met if the proprietor ensures that–
  - 32(1)(d) following an inspection under section 108 or 109 of the 2008 Act, a copy of the report of the inspection (if it has been sent to the proprietor) is published and maintained on the school's internet website, and provided to the parents of each registered pupil, by any date specified by the body who conducted the inspection
- 32(2) The information specified in this sub-paragraph is–
  - 32(2)(b) either–
    - 32(2)(b)(i) where the proprietor is an individual, the proprietor's full name, address for correspondence during both term-time and holidays and a telephone number or numbers on which the proprietor may be contacted, or
    - 32(2)(c) where there is a governing body, the name and address for correspondence of its Chair
- 32(3) The information specified in this sub-paragraph is–
  - 32(3)(g) a copy of the report of any inspection carried out under sections 108 or 109 of the 2008 Act or section 87(1) of the 1989 Act.

## The school now meets the following requirements of the independent school standards

*Standards that were not met at the previous inspection, but are now judged to be met at this inspection*

### Part 1. Quality of education provided

- 2(2) For the purposes of paragraph (2)(1)(a), the matters are–
  - 2(2)(d) personal, social, health and economic education which–
    - 2(2)(d)(i) reflects the school's aim and ethos; and
    - 2(2)(d)(ii) encourages respect for other people, paying particular regard to the protected characteristics set out in the 2010 Act
  - 2(2)(e) for pupils receiving secondary education, access to accurate, up-to-date careers guidance that–
    - 2(2)(e)(i) is presented in an impartial manner

- 2(2)(e)(ii) enables them to make informed choices about a broad range of career options; and
- 2(2)(e)(iii) helps to encourage them to fulfil their potential
- 2(2)(h) that all pupils have the opportunity to learn and make progress; and
- 2(2)(i) effective preparation of pupils for the opportunities, responsibilities and experiences of life in British society.
- 3 The standard in this paragraph is met if the proprietor ensures that the teaching at the school–
  - 3(a) enables pupils to acquire new knowledge and make good progress according to their ability so that they increase their understanding and develop their skills in the subjects taught
  - 3(b) fosters in pupils self-motivation, the application of intellectual, physical and creative effort, interest in their work and the ability to think and learn for themselves
  - 3(c) involves well planned lessons and effective teaching methods, activities and management of class time
  - 3(d) shows a good understanding of the aptitudes, needs and prior attainments of the pupils, and ensures that these are taken into account in the planning of lessons
  - 3(f) utilises effectively classroom resources of a good quality, quantity and range
  - 3(g) demonstrates that a framework is in place to assess pupils’ work regularly and thoroughly and use information from that assessment to plan teaching so that pupils can progress.

## **Part 2. Spiritual, moral, social and cultural development of pupils**

- 5 The standard about the spiritual, moral, social and cultural development of pupils at the school is met if the proprietor–
  - 5(b) ensures that principles are actively promoted which–
  - 5(b)(vi) encourage respect for other people, paying particular regard to the protected characteristics set out in the [Equality] Act 2010.

## **Part 3. Welfare, health and safety of pupils**

- 15 The standard in this paragraph is met if the proprietor ensures that an admission and attendance register is maintained in accordance with the Education (Pupil Registration) (England) Regulations 2006.

## **Part 4. Suitability of staff, supply staff and proprietors**

- 18(2) The standard in this paragraph is met if–
- 18(2)(a) no such person is barred from regulated activity relating to children in accordance with section 3(2) of the 2006 Act where that person is or will be engaging in activity which is regulated activity within the meaning of Part 1 of Schedule 4 to that Act
- 18(2)(c) the proprietor carries out appropriate checks to confirm in respect of each such person–
  - 18(2)(c)(i) the person’s identity
  - 18(2)(c)(iii) the person’s right to work in the United Kingdom; and
  - 18(2)(c)(iv) where appropriate, the person’s qualifications

- 18(2)(d) the proprietor ensures that, where relevant to any such person, an enhanced criminal record check is made in respect of that person and an enhanced criminal record certificate is obtained before or as soon as practicable after that person's appointment
- 18(2)(e) in the case of any person for whom, by reason of that person living or having lived outside the United Kingdom, obtaining such a certificate is not sufficient to establish the person's suitability to work in a school, such further checks are made as the proprietor considers appropriate, having regard to any guidance issued by the Secretary of State
- 21(3) The information referred to in this sub-paragraph is–
- 21(3)(a) in relation to each member of staff ("S") appointed on or after 1st May 2007, whether–
- 21(3)(a)(i) S's identity was checked
- 21(3)(a)(ii) a check was made to establish whether S is barred from regulated activity relating to children in accordance with section 3(2) of the 2006 Act
- 21(3)(a)(iv) checks were made to ensure, where appropriate, that S had the relevant qualifications
- 21(3)(a)(v) an enhanced criminal record certificate was obtained in respect of S
- 21(3)(a)(vi) checks were made pursuant to paragraph 18(2)(d)
- 21(3)(a)(vii) a check of S's right to work in the United Kingdom was made; and
- 21(3)(a)(viii) checks were made pursuant to paragraph 18(2)(e).
- 21(4) The information referred to in this sub-paragraph is, in relation to each member of staff in post on 1st August 2007 who was appointed at any time before 1st May 2007, whether each check referred to in sub-paragraph (3) was made and whether an enhanced criminal record certificate was obtained, together with the date on which any check was completed or certificate obtained.
- 21(6) The information referred to in this sub-paragraph is, in relation each member ("MB") of a body of persons named as the proprietor appointed on or after 1st May 2007, whether the checks referred to in paragraph 20(6)(b) were made, the date they were made and the date on which the resulting certificate was obtained.
- 21(7) The information referred to in this sub-paragraph is, in relation to each member of a body of persons named as the proprietor in post on 1st August 2007 who was appointed at any time before 1st May 2007–
- 21(7)(a) whether each check referred to in sub-paragraph (6) was made; and
- 21(7)(b) whether an enhanced criminal record certificate was obtained, together with the date on which any check was completed or certificate obtained.

## **Part 5. Premises of and accommodation at schools**

- 27 The standard in this paragraph is met if the proprietor ensures that–
- 27(b) external lighting is provided in order to ensure that people can safely enter and leave the school premises.
- 29(1) The standard in this paragraph is met if the proprietor ensures that suitable outdoor space is provided in order to enable–
- 29(1)(a) physical education to be provided to pupils in accordance with the school curriculum.

Any complaints about the inspection or the report should be made following the procedures set out in the guidance 'Raising concerns and making a complaint about Ofsted', which is available from Ofsted's website: [www.gov.uk/government/publications/complaints-about-ofsted](http://www.gov.uk/government/publications/complaints-about-ofsted). If you would like Ofsted to send you a copy of the guidance, please telephone 0300 123 1231, or email [enquiries@ofsted.gov.uk](mailto:enquiries@ofsted.gov.uk).

### Parent View

You can use Parent View to give Ofsted your opinion on your child's school. Ofsted will use the information parents and carers provide when deciding which schools to inspect and when and as part of the inspection.

You can also use Parent View to find out what other parents and carers think about schools in England. You can visit [www.parentview.ofsted.gov.uk](http://www.parentview.ofsted.gov.uk), or look for the link on the main Ofsted website: [www.gov.uk/ofsted](http://www.gov.uk/ofsted).

The Office for Standards in Education, Children's Services and Skills (Ofsted) regulates and inspects to achieve excellence in the care of children and young people, and in education and skills for learners of all ages. It regulates and inspects childcare and children's social care, and inspects the Children and Family Court Advisory and Support Service (Cafcass), schools, colleges, initial teacher training, further education and skills, adult and community learning, and education and training in prisons and other secure establishments. It assesses council children's services, and inspects services for children looked after, safeguarding and child protection.

If you would like a copy of this document in a different format, such as large print or Braille, please telephone 0300 123 1231, or email [enquiries@ofsted.gov.uk](mailto:enquiries@ofsted.gov.uk).

You may reuse this information (not including logos) free of charge in any format or medium, under the terms of the Open Government Licence. To view this licence, visit [www.nationalarchives.gov.uk/doc/open-government-licence](http://www.nationalarchives.gov.uk/doc/open-government-licence), write to the Information Policy Team, The National Archives, Kew, London TW9 4DU, or email: [psi@nationalarchives.gsi.gov.uk](mailto:psi@nationalarchives.gsi.gov.uk).

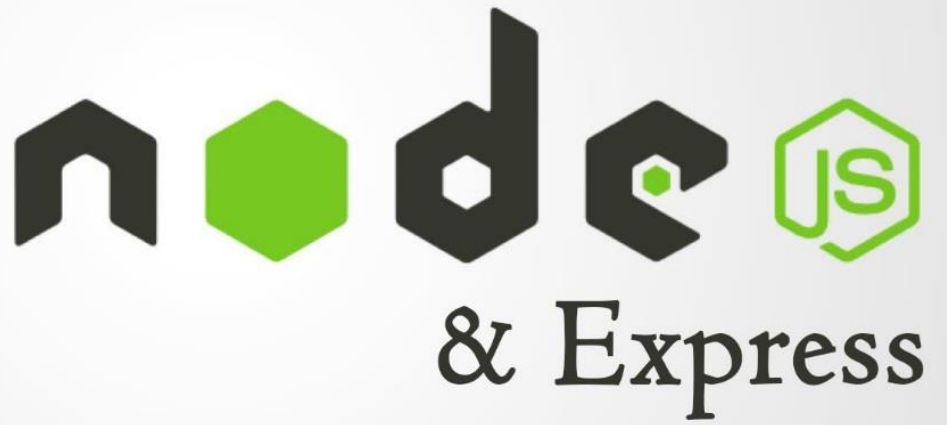
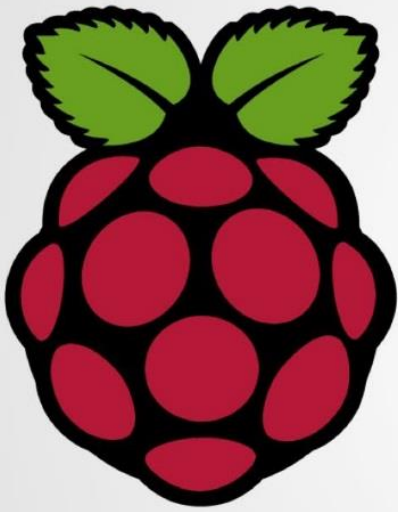
This publication is available at <http://reports.ofsted.gov.uk/>.

Interested in our work? You can subscribe to our monthly newsletter for more information and updates: <http://eepurl.com/iTrDn>.

Piccadilly Gate  
Store Street  
Manchester  
M1 2WD

T: 0300 123 1231  
Textphone: 0161 618 8524  
E: [enquiries@ofsted.gov.uk](mailto:enquiries@ofsted.gov.uk)  
W: [www.gov.uk/ofsted](http://www.gov.uk/ofsted)

© Crown copyright 2022



تالیف: [حسن تفرشی](#)



صفحه	فهرست
3	درباره کتاب
3	ملاحظات جهت آگاهی خوانندگان
5	ساختار
6	جاوا اسکریپت و Node.js
6	جاوا اسکریپت و شما
7	جاوا اسکریپت سمت سرور
8	ساختن یک اپلیکشین کامل بوسیله Node.js
8	پشته اپلیکشین
9	ساخت یک پشته اپلیکشین
10	آنالیز کد HTTP سرور
12	چگونه تابع سرویس https را پاس می دهد
14	Node.js و express
14	اجرا کردن پروژه با express
15	یک اپ ساده با express
18	express و نمایش
19	JADE چیست
19	JADE و HTML
20	JADE و NODE.JS
20	ساخت یک اپ ساده با express
23	یک دایرکتوری عمومی برای اپ
27	مولد خودکار app در express
29	قدرت دادن express با خاصیت میان افزاری
35	Express قدرت گرفته از ماژول Node
38	درخواست log به اپ
40	استفاده از یک فایل پیکربندی شده
41	Set و Get اپلیکشین
42	Environment متفاوت اکسپرس
43	منابع

## درباره کتاب :

این کتاب با هدف آشنایی شما با node.js و درباره مواردی که احتیاج دارید درباره توسعه اپلیکشن Node.js نیاز دارد تالیف شده است و این متن مشابه آموزش های “Hello World” می باشد .

نکته: کدهای در این کتاب در نسخه های ۰.۸.۸ به بالا Node.js به درستی کار می کند

## ملاحظات جهت آگاهی خوانندگان :

این کتاب مناسب افرادی است که پیشتر از خواندن این کتاب آشنای کافی با زبانی object-oriented نظیر PHP , Python , Ruby یا Java و همچنین حداقل تجربه ی در زمینه کار با JavaScript را داشته باشند و اگر پیشتر از این هیچ آشنائی با Node.js ندارید این کتاب مناسب شما می باشد .

لازم است خواننده پیشتر از این در جهت توسعه با زبان های یاد شده بالا تجربه ای داشته باشد و همچنین با ساختار های Object-oriented آشنای داشته باشد زیرا این متن به توضیح جنس متغیر ها یا کنترل ساختمان و توابع مسائل نظیر به این نمی پردازد . جهت درک متن این کتاب نیاز است شما با ساختارهای یاد شده پیشتر از این آشنائی داشته باشید .

به هر حال همانطور که مستحضر هستید ساختار توابع (functions) و شی (object) در جاوا اسکریپت با اکثر زبان ها متفاوت است و در متن پیش رو در این خصوص جزئیات بیشتری ارائه خواهد شد.

## ساختار

هدف این است که خواننده عزیز بعد از اتمام این کتاب توانی خلق یا به عبارت دیگری تولید یا به عبارت عامیانه توانی کدنویسی کامل یک اپلیکشین تحت که به کاربرها این اپلیکشین این اجازه را می دهد که فایل ها خود را در این اپلیکشین تحت وب بارگزاری کنند و در آموزش قدم به قدم تا رسیدن به خلق کامل این اپلیکشن به توضیح قسمت به قسمت کدها می شود .

حقیقتا همانطور که شما فکر می کنید قرار نیست این اپلیکیشن ساده دنیا را تغییر دهد اما ما همانطور که در پارگراف بالا یاد شده است این فقط کد نوشتن نیست ولی برای شروع کافی است . شما در طول این کتاب خواهید دید این اپلیکشین ساده تحت وب چه چیزهای در خصوص ساختار framework برای شما در Node.js را روشن خواهد کرد.

پیشتر از شروع می ببینیم که تفاوت توسعه دادن جاوا اسکریپت در Node.js با توسعه دادن جاوا اسکریپت در یک مرورگر (Browser) .

در ادامه ادامه به آموزش روش قدیمی و قدم به قدم به نام "Hello World" که بسیار پرثمره است خواهیم پرداخت و همچنین به معرفی Express می پردازم .

## جاوا اسکریپت و Node.js

### جاوا اسکریپت و شما

اگر شما هم مثل من توسعه را از طریق HTML را سال های پیش شروع کردید شما در سالهای قبل با زبانی آشنا شدید که به آن جاوا اسکریپت می گفتند آشنا شدید که کارهای بسیار باحال و دوست داشتنی انجام می داد و همچنان به این وظیفه خود ادامه می دهد اما شما فقط برای مسائل خیلی مقدماتی و تعاملی در توسعه وب ها استفاده می کردید .

حتمی شما نیز می خواستید یک حرکت واقعی یا بصورت دقیق یک توسعه واقعی با جاوا اسکریپت انجام دهید و احتمالاً می خواستید بدانید چگونه می توان یک وب سایت های پیچیده را از این طریق ساخت و همانطور که شما کد نویسی و توسعه با زبان های نظیر جاوا و پی اچ پی و ... را شروع و پایان رساندید . گوشه چشمی همچنان در جهت توسعه به JavaScript داشته اید و شما دید به وسیله جی کوئری<sup>۱</sup> بصورت پیشرفته تر از جاوا اسکریپت می توان عمل کرد اما این ساختار در حقیقت بیشتر درباره window.open() می باشد.

به هر حال در زمان استفاده جی کوئری در پروژه ها همه چیز خوب است اما در پایان پروژه شما همچنان یک کاربر هستید که از جاوا اسکریپت استفاده می کنید نه یک توسعه دهند جاوا اسکریپت<sup>۲</sup>

الان که Node.js در سمت سرور در اختیار می باشد . به نظر شما باحال نیست ؟ جاوا اسکریپت سمت سرور!

همه چیز کامپیوتر و آی تی به زمان بر می گردد و بروز شدن ، همانطور که شما دنبال بروز کردن اطلاعات هستید باید بدانید ، نوشتن اپلیکیشن Node.js یک داستان است و متوجه بودن این امر که چرا شما باید با جاوا اسکریپت اپلیکیشن بنویسید داستان متفاوتی است .

طریقه نوشتن اپلیکیشن های Node.js به شکلی می باشد که شما فقط احساس نمی کنید که در حال استفاده جاوا اسکریپت هستید در حقیقت شما در حال توسعه دادن Node.js می باشید و به عبارت صحیح تر شما توسعه دهنده هستید نه یک کاربر.

---

<sup>۱</sup> JQuery

<sup>۲</sup> JavaScript Developer

در اینترنت منابعی مختلفی در خصوص Node.js و جاوا اسکریپت وجود دارد اما در بسیار موارد داکيومنت و متون به تنهایی کافی نیستند و در حقیقت شما در یادگیری مسئله ای نیاز به راهنما دارید .

## جاوا اسکریپت سمت سرور

از زمان پیدایش جاوا اسکریپت تا به امروز در سمت مرورگر قابلیت اجرا داشته است اما این فقط محتوا است , جاوا اسکریپت یک زبان کامل است. این متن به شرح این موضوع می پردازد که چگونه شما می توانید به تعریف جاوا اسکریپت سمت سرور بپردازید. Node.js به شما اجازه می دهد جاوا اسکریپت را در خارج یک مرورگر اجرا کنید.

Node.js دو موضوع است اول : یک محیط زمان اجرا و یک کتابخانه <sup>۳</sup> .

شما نیاز دارید که Node.js را نصب کنید در خصوص نصب به [اینجا](#) کلیک کنید

## "Hello World"

خوب حالا بریم سراغ اولین آپ که با Node.js خواهیم نوشت این آپ سلام دنیا یا عبارت معروف " Hello world " نام دارد .

ایدتور مورد علاقه خود را باز کنید و فایل با نام helloworld.js بسازید .

قصد داریم از طریق STDOUT عبارت "Hello world" را برگردانیم یا به عبارت بهتر آن را چاپ کنیم . کد زیر را در فایل ذخیره کنید

```
console.log("Hello World");
```

حالا کد بالا را از طریق Node.js با فرمان زیر اجرا کنید :

```
node helloworld.js
```

شما در خروجی حتمی باید Hello world را مشاهده کنید. من هم با شما موافقم این آپ بسیار ساده و خسته کننده است حالا بیاد یک چیز یا به عبارت تخصص یک آپ واقعی تری بنوسیم .

---

<sup>3</sup> library

## ساختن یک اپلیکشین کامل بوسیله Node.js

### به سراغ اپلیکشین آپلود برویم

شما با جستجو در گوگل و دیدن سورس های دیگر نیز می توانید به این هدف که ساختن یک اپلیکیشن آپلود هست برسید اما این آن چیزی نیست که ما در اینجا قصد داریم به آن برسیم . هدف ما فهمیدن ساختار و روش کار کردن با node.js می باشد.

### پشته /پلیکشین

- برای اجرا وب پیج ها به ساختار HTTP سرور نیاز داریم
  - سرور باید توانای پاسخ گوئی در خواست های متفاوت از آدرس های متفاوت را داشت باشد .
  - ما در حقیقت به request handler در سرور نیاز داریم در جهت اجرا درخواست های وارده به سرور .
  - سرور باید توانای پشتیبانی از متد دیتا پست (Post) جهت هندل کردن درخواست ها داشته باشد در حقیقت نیاز به یک request data handling داریم
- پیشتر یک لحظه فکر کنیم چگونه می توان یک پشته با php ساخت . این موضوع یک راز نیست و با نصب آپاچی سرور این امر با کمک mod\_php5 قابل انجام است.
- در حقیقت ما نیاز داریم که سرور توانای دریافت درخواست های http از صفحات وب را داشته باشد . این امر ممکن نمی شود بوسیله خود php .
- بوسیله node.js مسئله کمی متفاوت است . ما فقط به اجرا اپلیکشین نمی پردازیم بلکه به اجرا تمامی HTTP سرور نیز می پردازیم . در این خصوص اپلیکشن وب ما و وب سرور در بصورت پایه ای یکی می شوند .
- ممکن است به نظر بیاد برای عملی شدن امر بالا کارهای زیاد باید انجام داد اما ما خواهیم دید بوسیله node.js یک لحظه بیشتر طول نمی کشد .

خوب بریم سراغ اجرا قسمت اول پشته در HTTP سرور.

## ساخت یک پشته /پلیکشین

در سمت سرور

اول فایل اصلی که می خواهیم اپلیکشین ما را شروع کنید و یک ماژول فایل که قرار است کد HTTP سرور در آن قرار گیر را می سازیم.  
پیشنهاد می شود بصورت استاندارد اسم فایل اصلی را `index.js` بگذارید و اسم فایل ماژول سرور را `server.js` انتخاب کنید.  
فایل `server.js` ایجاد شده را در مسیر اصلی دایرکتوری پروژه خود قرار دهید و کد زیر را در درون آن قرار دهید :

```
var http = require("http");  
  
http.createServer(function(request, response) {  
  response.writeHead(200, {"Content-Type":  
    "text/plain"});  
  response.write("Hello World");  
  response.end();  
}).listen(8888);
```

به همین سادگی ! شما یک HTTP سرور ساختید . حالا برای اثبات صحت کار آن را اجرا و تست کنید . اول , اسکریپت اجرای `node.js` را اجرا کنید :

`node server.js`

حالا مرورگر خود را باز کنید و آدرس <http://localhost:8888/> را وارد کنید شما خواهید دید که

نوشته "Hello World" به نمایش در آمده است

جالب بود نه ؟ حالا صحبتی که مطرح می شود این است که چگونه پروژه های آینده خود را سازماندهی کنیم .حتمی به این موضوع خواهیم پرداخت در فصول آینده .

## آنالیز کد HTTP سرور

حالا وقت اینه که ببینم دقیقا چه اتفاقی در کد قسمت قبل افتاده است . خط اول درخواست یک ماژول از جنس http شده بوسیله node.js و نام این متغیر را http گذاشته ایم . در خط بعد یکی از توابع ماژول http را فراخوانی کردیم که اسم آن تابع createServer می باشد این تابع یک شئی (object) را برای ما در دسترس قرار می دهد این شئی شامل یک متد به نام listen می باشد . متد listen یک مقدار عددی را بعنوان وردی می گیرد که همان شماره پورت برای http سرور ما جهت منتظر بودن فرمان می باشد.

برای لحظه ای توضیحات قسمت را در پس زمینه ذهنتان حفظ کنید و قسمت http.createServer توجه داشته باشید .

ما می توانستیم کد شروع گرفتن سرویس را بنویسم و منتظر پاسخ از پرت ۸۸۸ باشیم مانند کد زیر

```
var http = require("http");
var server = http.createServer();
server.listen(8888);
```

این کد می تواند یک سرویس http را شروع و منتظر پاسخ از پورت 8888 باشد و هیچ کار دیگری نیز انجام ندهد ( نه زمانی که پاسخی یا درخواستی دریافت می کند )

کد اول خیلی بهتر و جالب تر است ( و اگر شما در پس زمینه از بیشتر یک زبان مانند php نیز استفاده کرده باشید ) . قسمتی از تابع را تعریف کرده اید (در کد اول) که شما از پارمتر اولی که در createServer() را فراخوانی کرده اید

نکته دیگر کد اول این است که ما فقط پارمترهای را به createServer() داده ایم و همان را نیز فراخوانده ایم . دلیل این موضوع این است که توابع جاوا اسکریپت می تواند توابع را مثل مقادیر به یک دیگر پاس دهند ( برای اطلاعات بیشتر در خصوص رفتار و توابع در زبان های مختلف به کتاب "Concepts of Programming Languages 10th Edition" مراجعه کنید) .

برای مثال

```
function say(word) {
```



---

```
console.log(word);
}
function execute(someFunction, value) {
  someFunction(value);
}
execute(say, "Hello")
```

به دقت به کد بالا نگاه کنید. اتفاقی که در کد بالا می افتد این است که تابع `say` را به صورت پارمتر به عنوان پارمتر اول تابع `execute` ارسال می شود. مقدار بازگشتی تابع `say` را به عنوان پارمتر تابع `execute` نشده است بلکه خود تابع `say` به عنوان یک پارمتر ارسال شده است. تابع `say` به متغیر محلی `someFunction` تابع `execute` پاس داده می شود (اگر در خصوص پاس دادن مقادیر و متغیرها محلی و توابع آشنای کافی ندارید به کتاب شماره یک در قسمت منابع مراجعه کنید) و تابع `execute` می تواند تابع موجود در متغیر `somefunction()` را صدا بزند. (در جهت تعریف یک تابع به عنوان یک متغیر ورودی تابع دیگر حتمی پرانتز گذاشته شود که تابع ی که در حال تعریف آن هستید متوجه شود که مقدار متغیر دریافتی از جنس تابع می باشد)

البته مشخص است که تابع `say` به دلیل اینکه یک پارمتر بیشتر ندارد، `execute` می تواند یک متغیر را زمان صدا زدن `somefunction` دریافت کنید.

می توان، همانطور که مشاهده کردید یک تابع یک پارمتری را به تابع دیگری بوسیله نام آن تابع پاس دهید و این توضیح به این شکل صادق است که می توان یک تابع را بصورت پارمتری در تابع دیگر تعریف کرد بصورت مثال:

```
function execute(someFunction, value) {
  someFunction(value);
}
execute(function(word) { console.log(word) }, "Hello");
```

در کد فوق یک تابع را به عنوان تابع پارمتری دیگری تعریف کرده ایم. این یکی از ویژگی های جالب است که جاوا اسکریپت را متمایز از دیگر زبان ها.

## چگونه تابع سرویس **https** را پاس می دهد

با توجه به اطلاعات کسب شده به عقب برگردیم به سراغ سرویس **http** :

```
var http = require("http");
http.createServer(function(request, response) {
  response.writeHead(200, {"Content-Type":
    "text/plain"});
  response.write("Hello World");
  response.end();
}).listen(8888);
```

اکنون در کد بالا مشخص است که چه اتفاقی برای **createServer** افتاده است .

حالا بررسی می کنیم که **node.js** چگونه کد ما بصورت **run time** اجرا می کند روش اجرا **run time** روش مختص **node.js** نمی باشد اما با زبان های دیگر مثل **PHP** , **Ruby** , **Python** یا جاوا متفاوت است

کد ساده زیر را مشاهده کنید :

```
var result = database.query("SELECT * FROM hugetable");
console.log("Hello World");
```

در حال حاضر قصد نداریم درباره اتصال به دیتا بیس صحبت کنیم بلکه فقط کد فوق جهت یک مثال ساده در بالا قرار گرفته است .

در اول مرحله اول کوئری از دیتا بیس ردیف های زیادی از دیتا بیس بر می گرداند و در خط دوم عبارت **"Hello world"** در کنسول قرار می گیرد. خواندن این همه ردیف از دیتا بیس ( منظور دیتا بیسی با چندین میلیون رکورد می باشد ) بسیار ترسناک است و چندین ثانیه طول می کشد .

به این روشی که کد را نوشتیم در جاوا اسکریپت مترجم **node.js** اول تمام مقدار برگشتی از دیتا بیس را می خواند و سپس می تواند تابع **console.log()** را اجرا کند .

اگر تکه کد بالا از طریق زبان php نوشته شده بود به روش یاد شده بالا کار می کرد. اول همه نتایج را می خواند و بعد خط بعدی از کد را اجرا می کرد و اگر این کد یک اسکریپت از صفحه وب بود کاربر باید چندین دقیقه برای لود شدن صفحه صبر می کرد .

به هر حال در روش مدل php این به یک مشکل بزرگ تبدیل نمی شود و وب سرور شروع به درخواست پروسس های php از هر درخواست دریافتی از http می کند . اگر یکی از نتایج درخواست های ارسالی در حال اجرا یکی از کد های مشابه بالا باشد خروجی این درخواست برای کاربر منحصر می باشد اما در خواست های کاربران دیگر برای صفحات دیگر را تحت تاثیر قرار نمی دهد .

نحوه اجرا این مدل در node.js متفاوت است در آنجا فقط یک پروسه وجود دارد و اگر یک درخواست به کد بالا داشته باشیم که باعث گرفتن کوئری با سرعت کم می شود در قسمتی از فرآیند این موضوع تمام پروسه های دریافتی را تا تمام شدن این کوئری تحت تاثیر قرار می دهد.

ما می دانیم با آنالیز این محتوا و دوباره نوشتن این کد مشکل حل خواهد شد به کد زیر توجه کنید

```
database.query("SELECT * FROM hugetable",  
function(rows) {  
var result = rows;  
});  
console.log("Hello World");
```

همانطور که در کد بالا مشاهده می کنید database.query() مقدار بازگشتی را به پارمتر دوم که بصورت تابع تعریف شده است پاس می دهد .

در حالت قبلی این کد اول کد کوئری دیتابیس را اجرا می کرد و بعد از خاتمه کنسول نوشتن اجرا می شد. حالا node.js می تواند با هندل درخواست ناهنگام از دیتا بیس

در متن بعدی مفصلا در خصوص دیتابیس و نودجی اس بحث خواهیم کرد

## *express و Node.js*

### *اجرا کردن پروژه با express*

برای اجرا کردن تمامی پروژه های که با Express می سازیم رویه زیر دنبال کنید :

۱- فایلی با نام package.json<sup>۴</sup> با محتوای زیر در شاخه ای مورد نظر بسازید ( بصورت مثال cd express-app \$ در ریشه مورد علاقه )

```
{  
  "name": "test-app",  
  "version": "0.0.1",  
  "private": true,  
  "scripts": {  
    "start": "node app"  
  },  
  "dependencies": {  
    "express": "3.2.6",  
    "jade": "*"   
  }  
}
```

---

<sup>4</sup> [JavaScript Object Notation](#)

توضیح فیلد های استفاده شده در بالا به شرح زیر می باشد

فیلد	توضیح
Name	نام انتخابی شما برای ماژولی که قصد ساختن آن را دارید
Version	نسخه ماژول را از اینجا معرفی می کنید
Private	اجازه پابلیک بودن یا نبودن آن در ریجستری npm از این طریق تنظیم می شود که در این پروژه مقدار true دارد که تعیین می کند اپلیکیشن بصورت شخصی (خصوصی) می باشد
Scripts	دستورات اجرای npm در پروژه بالا بصورت ویژه دستور node app فقط دستور اجرای ماژول می باشد .
Dependencies	ماژول های وابسته در اینجا معرفی می کنید که در بالا ماژول های express و jade بصورت ویژه معرفی شده اند.

بعد ساختن فایل package.json در هر شاخه ی که فایل مورد نظر ساخته شده است دستور زیر را در CMD اجرا نمایید

**C:\express-app\ npm install**

**یک اپ ساده با express**

خوب اپ ساده زیر با نام app.js در شاخه ی که در بالا ساختید با محتوای زیر ذخیره کنید و دستور node app را در همان شاخه در CMD اجرا نمایید .

```
// کتابخانه http
var http = require('http');
// ماژول express
var express = require('express');
```

```
// express یک نمونه ساخته شده از
var app = express();
// شروع اپ
http.createServer(app).listen(3000, function() {
  console.log('Express app started');
});
// یک روت برای صفحه خانه
app.get('/', function(req, res) {
  res.send('Welcome!');
});
```

برای اجرا اپ بالا دستور `node app` را اجرا کنید . بعد از اجرا پیام **“Express app started”** مشاهده خواهید کرد و برای پایان دادن به اجرا سرور دکمه `ctrl+c` را کلیک کنید .

نکته: در مورد پیام خطا 404 و ۵۰۰ و هندل های آن در ادامه توضیح خواهد داده می شود

### آنالیز خروجی:

کلمه `welcome` در آدرس <http://localhost:3000> در صفحه اصلی نمایش داده می شود و در دیگر درخواست های مقدار پیام خطا ۴۰۴ را بر می گرداند.



اگر سورس صفحه را نگاه کنید می بینید که یک پاسخ بصورت تکس دریافت کرده اید . سوالی که الان مطرح است برای پاسخ بصورت html چه باید کرد ؟!

خوب بصورت نمونه خط `res.send('Welcome!')` مقدار متنی را به `<h1>Welcome!</h1>` تغییر دهید و سرور رو ری استارت کنید و صفحه را در مرورگر رفرش کنید و خروجی مانند تصویر زیر را خواهید دید .



# Welcome!

نکته : برای دیدن خروجی جدید حتمی با سرور را restart کنید و یا اینکه از supervisor استفاده کنید با کمک این مازول بعد از هرگونه تغییر سرور را بصورت اتوماتیک restart می کنید برای اطلاعات بیشتر درباره supervisor و آموزش استفاده به اینجا <https://github.com/isaacs/node-supervisor> مراجعه کنید.

حالا چگونگی ارسال یک صفحه html را دنبال خواهیم کرد . اپ های Express کامپونت ویژه ای دارند به نام views دارد. هر جا که شما لازم داشته باشید از html بصورت زبان قالب استفاده کنید هر تغییری در فرم html هسته app نیاز باشد بوسیله views اعمال و بصورت html در خروجی بدون نیاز به restart کردن سرور تغییر و ذخیره خواهد کرد.

## *express و نمایش :*

با محوریت نمایش محتوا اپی می نویسم . خوب برای این منظور شاخه ای باید ایجاد شود . همیشه این موضوع را در نظر بگیرد نام گذاری ها به شکلی باشد که با محتوای آن در ارتباط باشد این موضوع ساده به بهتره و قابل درک بودن اپ شما کمک می کند.

شاخه ای با نام **views** در مسیر مورد نظر بسازید یا با دستور **mkdir views** در ترمینال سیستم عامل خود.

حالا در شاخه **views** دو فایل با نام های **index.jade** برای نمایش صفحه خانه و **hello.jade** برای صفحه سلام بسازید . اگر شما یک برنامه نویس باشید حتمی الان کنجکاو شده باشید چرا پسوند **JADE** و یا اصلن این پسوند چیست ؟

فایل **index.jade** در مسیر **views** بسازید و محتوای زیر را در درون آن قرار دهید

```
doctype html
html(lang="fa" dir="rtl")
  head
    title= page

  body
    h1 جی-د سازنده قالب

    h3 شما شگفت انگیز هستی
    p پیش به سوی آینده
    p.
```

جی-د یک زبان سازنده قالب که در ساخت آن کارایی و سادگی در نظر گرفته شده است



فایل hello.jade در مسیر views بسازید و محتوای زیر را در درون آن قرار دهید

```
html(lang="fa" dir="rtl")
```

```
head
```

```
title Hello
```

```
body
```

```
b Hello!
```

```
p سلام
```

## *JADE چیست*

[JADE](#) در معنی لغوی اسم نوع سنگ قیمتی و زینتی به نام [یشم سبز](#) می باشد اما JADE یا معادل پینگلیش آن جی-د در اینجا زبانی است با ساختار [شی گرا](#)<sup>۵</sup> که اولین بار در مرکز نیوزلند بنیان گذاری شد و اولین نسخه آن در سال ۱۹۹۶ ساخت شد .

براساس ساختار end-to-end طراحی شده است و همچنین شامل API برای زبان های دیگر نیز می شود شامل .Net Framework و JAVA و C/C++ و وب سرورس ها .

## *JADE و HTML*

جی-د از ساختارهای متفاوتی پشتیبانی می کند اما برای دور نشدن از مبحث بصورت ویژه در خصوص پشتیبانی آن از HTML توضیح می دهیم.

جی-د خیلی ساده تر از ASP.NET در خصوص خلق و توسعه قالب های HTML و صفحات کار می کند و زمانی که یک قالب برای صفحه HTML طراحی می شود بصورت چند بخشی طراحی می شود و بر خلاف ساختار کلی جی-د در زمان کار با ساختار HTML از متد Front End استفاده می کند .

---

<sup>5</sup> Object-oriented

## NODE.JS و JADE

جی-د در نود یک موتور ساخت قالب می باشد و رندر پیش فرض در فریم ورک express می باشد و به همین دلیل از این زبان در طول این متن استفاده خواهد شد و در صورت امکان از متدهای دیگر نیز مثالی زده خواهد شد .

برای آشنای بیشتر با این ساختار به منابع زیر می توانید مراجعه کنید :

۱- <http://jade-lang.com/>

۲- [www.franz-enzenhofer.com](http://www.franz-enzenhofer.com)

۳- تبدیل html به jade : [html2jade.aaron-powell.com](http://html2jade.aaron-powell.com)

### ساخت یک اپ ساده با express

برای شروع یک شاخه جدید در مسیر مورد نظر خود بسازید که در اینجا ما در مسیر express-app شاخه ۱ را ساخت ایم و فایل به نام app.js که محتوای زیر در درون آن قرار گرفته است( فایل های این اپ بصورت پیوست قرار دارد )

اپ شماره ۱ :

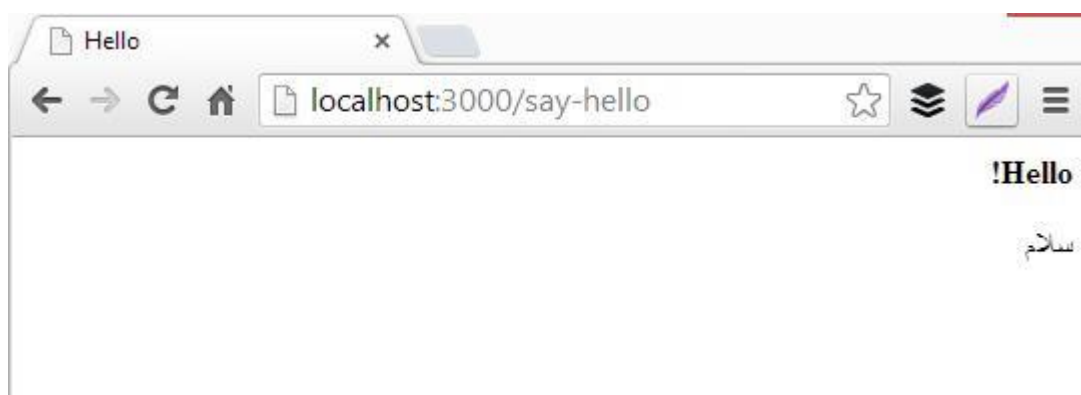
```
var http = require('http');
var express = require('express');
var app = express();
// موتور نمایش قالب را مشخص می کند
app.set('view engine', 'jade');
// مسیر نمایش را مشخص میکند
app.set('views', './views');
// صفحه اصلی با کمک تابع زیر تنظیم می شود صفحه
// index
// از این طریق به عنوان صفحه اصلی انتخاب می شود
```

```
app.get('/', function(req, res) {
  res.render('index');
});
// masire say0hello az in tarigh be view render
mishavad
app.get('/say-hello', function(req, res) {
  res.render('hello');
});
// از این طریق زمانی که مسیر تست درخواست شود مقدار متنی
// this is a test
// نمایش داده می شود
app.get('/test', function(req, res) {
  res.send('this is a test');
});
// مانند تابع بالا عمل می کند زمانی که مسیر
//http://127.0.0.1:3000/jade
// درخواست می شود مقدار اچ تی ام ال نمایش داده می شود
app.get('/jade', function(req, res) {
  res.send("<h3>Jade is a terse and simple templating
  language .</h3>");
});
http.createServer(app).listen(3000, function() {
  console.log('App started');
});
```

اپ از طریق پرت ۳۰۰ اجرا و منتظر پاسخ و دریافت اطلاعات می ماند در اینجا بهتر است بگویم دریافت مسیر جدید خواهد بود. در زمان درخواست مسیر اصلی به آدرس: <http://127.0.0.1:3000/> محتوا صفحه Index.jade را نمایش می دهد (تصویر شماره ۱) و در هنگام درخواست مسیر <http://127.0.0.1:3000/say-hello> صفحه hello.jade به نمایش در می آید (تصویر شماره ۲) و همچنین موقع درخواست از مسیر <http://127.0.0.1:3000/jade> و <http://127.0.0.1:3000/test> محتوای که به آن در تابع مقدار دهی کردیم نمایش داده خواهد شد (تصویر شماره ۳ و ۴).



تصویر شماره ۱



تصویر شماره ۲



تصویر شماره ۳



تصویر شماره ۴

نکته : اگر سورس هر کدام از صفحه ها را نگاه کنید می بنید نمایش بصورت خروجی HTML انجام شده است و در صورت تغییر در هر کدام از صفحات jade نیاز به restart کردن اپ نمی باشد مگر اینکه در ساختار app.js تغییر ایجاد کنید.

### یک دایرکتوری عمومی برای اپ

هم اکنون چند قدم به داشتن توانای و دانش ساخت یک وب سایت تابعی و داینامیک نزدیکتر شدیم . حال به محتوا می خواهیم CSS , JS<sup>۶</sup> و تصاویر اضافه کنید و محل ذخیره سازی این فایل ها یک دایرکتوری عمومی خواهد بود .

Express از متد استاتیک<sup>۷</sup> middleware برای صدا زدن استفاده می کند

---

<sup>۶</sup> JavaScript

از روش زیر می توانید یک دایرکتوری را بعنوان منبع استاتیک تنظیم و یا به عبارت بهتر اضافه کنید

```
app.use(express.static('./public'));
```

و برای اضافه کردن چند دایرکتوری استاتیک در صورت نیاز می توانید بصورت زیر استفاده کنید

```
app.use(express.static('./public'));  
app.use(express.static('./files'));  
app.use(express.static('./downloads'));
```

حالا روش ساخت پروژه جدید را انجام دهید و بعد از آن دایرکتوری های زیر را بسازید

**\$ mkdir public**

**\$ mkdir public/images**

**\$ mkdir public/javascripts**

**\$ mkdir public/stylesheets**

این اسامی دایرکتوری جهت واضح و قابل درک بودن اپ انتخاب شده اند و شما می توانید در اپ های دیگر اسامی دلخواه خود را انتخاب کنید اما سعی کنید اسامی انتخاب کنید باعث پیچیدگی درک اپ برای خود و یا دیگر برنامه نویس ها نشود .

خوب یک تصویر به عنوان لوگو<sup>۸</sup> با نام logo.png در دایرکتوری images اضافه کنید من از لوگو [وب برو](#) استفاده کردم



یک فایل به نام main.js بسازید در دایرکتوری javascripts با محتوا زیر بسازید.

```
window.onload = function() {  
document.getElementById('smile').innerHTML = ':)';  
};
```

یک فایل دیگر به نام style.css در دایرکتوری stylesheets با محتوا زیر بسازید:

---

<sup>7</sup> [میان افزار](#)

<sup>8</sup> Logo = لوگو

---

```
#content {
  width: 220px;
  margin: 0 auto;
  text-align: center;
  border: 1px solid #ccc;
  box-shadow: 0 3px 4px #ccc;
  padding: 5px;
  font-family: Tahoma;
  font-weight: normal;
  color: black;
  font-size: 12px;
}
#smile
{
  -webkit-transform: rotate(-90deg);
  -moz-transform: rotate(-90deg);
  -o-transform: rotate(-90deg);
  font-family: sans-serif;
  font-size: 18px;
  font-weight: normal;
  color: orange;
}
```

فایل index.jade را به روز کنید

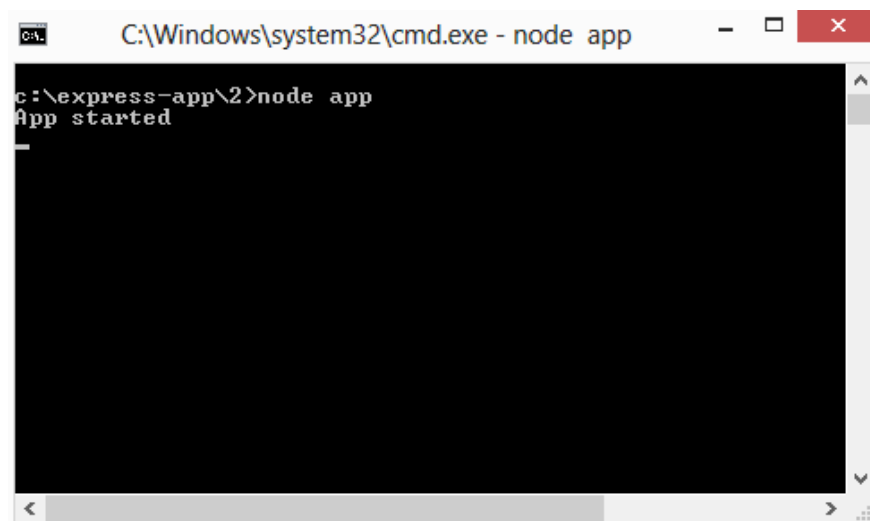
```
doctype html
html(lang="fa" dir="rtl")
title Welcome
script(src='javascripts/main.js')
link(rel='stylesheet', href='stylesheets/style.css')
body
#content
  img(src='images/logo.png')
  p خوش آمدید
  #smile
```

و کد زیر را در فایل app.js ذخیره کنید

```
var http = require('http');
var express = require('express');
var app = express();
```

```
app.set('view engine', 'jade');  
// دایرکتوری ها از طریق مشخص می شوند  
app.set('views', './views');  
// دایرکتوری  
//public  
// بصورت استاتیک از طریق زیر  
// برای دسترسی به اپ معرفی شده است  
app.use(express.static('./public'));  
// صفحه اصلی مشخص شده است  
app.get('/', function(req, res) {  
  res.render('index');  
});  
// سرور لوکال ساخته روی پورت ۳۰۰۰ منتظر دستور  
http.createServer(app).listen(3000, function() {  
  console.log('App started');  
});
```

اپ را اجرا کنید با دستور node : مانند تصویر زیر :



خوب خروجی مثل همیشه روی آدرس <http://localhost:3000> در دسترس می باشد مانند تصویر زیر





تبریک می گم اولین اپ با express خود را ساختید .

اپ شماره ۲

### مولد خودکار app در express

Express یک دستور خودکار جهت تولید اپ و دایرکتوری های views و ... غیر را دارد . جهت استفاده از این دستور و دیگر دستورات در ترمینال سیستم عامل خود تایپ کنید

#### ➤ express -h

تا مانند تصویر زیر این اطلاعات نمایش داده شود .

```
C:\Windows\system32\cmd.exe

c:\>express -h

Usage: express [options] [dir]

Options:

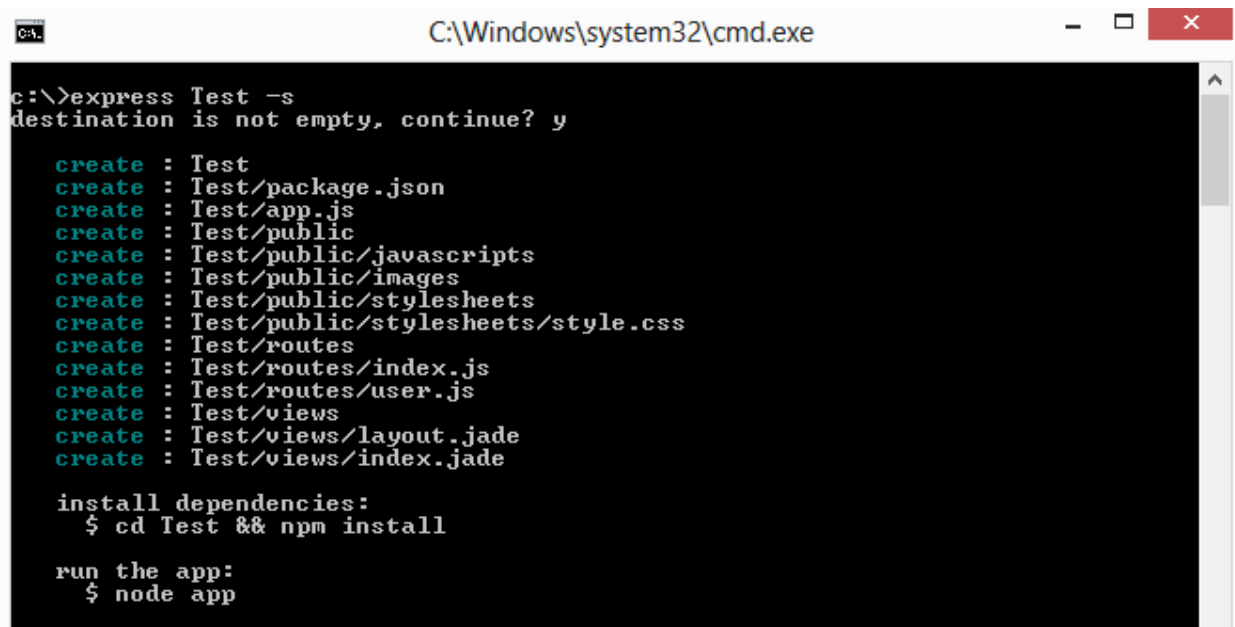
  -h, --help            output usage information
  -U, --version          output the version number
  -s, --sessions         add session support
  -e, --ejs              add ejs engine support (defaults to jade)
  -J, --jshtml           add jshtml engine support (defaults to jade)
  -H, --hogan            add hogan.js engine support
  -c, --css <engine>    add stylesheet <engine> support (less|stylus) (default
to plain css)
  -f, --force            force on non-empty directory
```

همانطور که در تصویر فوق می بینید روش استفاده توضیح داده شده است

روش کلی استفاده به این شرح است که اول دستور `express` و بعد فرمان انتخابی شما و نهایتن مسیر یا دایرکت که قرار است فرمان روی آن اعمال شود را وارد می کنید .

<b>-h</b>	فرمان کمک و نمایش همه فرمان ها
<b>-V</b>	نمایش نسخه <code>express</code>
<b>-s</b>	ساخت یک آپ
<b>-e</b>	پشتیبانی از موتور جی-د
<b>-J</b>	پشتیبانی از موتور <code>jshtml</code> (پیش فرض )
<b>-H</b>	اضافه کردن موتور <code>hogan.js</code> برای پشتیبانی (پیش فرض )
<b>-c</b>	اضافه کردن <code>CSS</code> (پیش فرض )
<b>-f</b>	نمایش محتوا در مسیر داده شده

بصورت نمونه روش استفاده از فرمان `express` در تصویر زیر نمایش داده شده است



```
c:\>express Test -s
destination is not empty, continue? y

create : Test
create : Test/package.json
create : Test/app.js
create : Test/public
create : Test/public/javascripts
create : Test/public/images
create : Test/public/stylesheets
create : Test/public/stylesheets/style.css
create : Test/routes
create : Test/routes/index.js
create : Test/routes/user.js
create : Test/views
create : Test/views/layout.jade
create : Test/views/index.jade

install dependencies:
$ cd Test && npm install

run the app:
$ node app
```

حالا آپ ساخته شده جدید را اجرا کنید و خروجی را مشاهده کنید به کمک دستور همیشگی :

## ➤ node app

همانطور که در تصویر فوق مشخص است تمام ساختار شبیه آنچه تا به الان می ساختیم بجز دایرکتوری routes و فایل های درون آن عجله نکنید به زودی درباره این دایرکتوری و فایل های موجود در آن نیز صحبت خواهیم کرد.

دیگر نیازی نیست برای هر پروژه فایل ها و دایرکتوری ها را ایجاد کنید با کمک خط فرمانی تمامی آنچه برای یک اپ نیاز دارید خلق می کنید .

نکته ی که درباره این مولد باید در نظر بگیرد تمام این محتوا تولید بصورت پیشنهاد می باشد و شما به دلخواه هر تغییری که بخواهید می توانید در آن ایجاد کنید .

## قدرت دادن express با خاصیت میان افزاری

به خاطر دارید که `app.use()` چگونه به عنوان یک میان افزار استفاده می کردیم . حالا به این موضوع فکر کنید که ما می توانیم میان افزار خود را کد نویسی کنیم 😊 حالا به استفاده میان افزارهای که به همراه expressJs است توجه کنید

توضیح	میان افزار
مسیر سیستمی اپ	router
درخواست لوگ به سرور	logger
پشتیبانی gzip/deflate در سرور	compress
تصدیق HTTP	basicAuth
پارس <sup>۹</sup> اپلیکیشن json	json
پارس اپلیکیشن از <sup>۱۰</sup> urlencoded	urlencoded
پارس چند بخشی / فرم دیتا	multipart
پارس کردن متن درخواست / در قالب JSON یا urlencoded و میان افزار چند بخشی	bodyParser
پایان زمانی درخواست	timeout

<sup>۹</sup> به عمل تجزیه و تحلیل کد توسط کامپایلر یا خواندن آن توسط انسان گفته می شود Parse

<sup>۱۰</sup> فراخوانی رمزنگاری (منظور متد ویا زبان مورد استفاده ) از مسیر تعریف شده شده را دارد : Urlencoded

cookieParser	پارس کننده <sup>۱۱</sup> کوکی
session	پشتیبانی از Session
cookieSession	کوکی بر پایه سشن
methodOverride	پشتیبانی از متد http
responseTime	نمایش مدت زمان پاسخگوی سرور
static	دایرکتوری ثابت و بارزش و اهمیت برای یک وب سایت
staticCache	نهان <sup>۱۲</sup> کردن برای میان افزار ثابت
directory	مسیر دایرکتوری ها
vhost	فعال کردن میزبانی مجازی <sup>۱۳</sup>
favicon	Favicon برای وب سایت
limit	محدود کردن سایز متن درخواندن
query	پارسر کردن کوئری GET
errorHandler	تولید یک پشته با فرمت HTML و دنبالی از پیام خطاهای سرور

در بالا لیستی از میان افزار های پیش فرض که در دسترس قرار دارد موجود می باشد و در زیر برای مثال از جگونگی به کار بستن یک میان افزار که در زیر از `responseTime` استفاده شده است را مشاهده می کنید

فایل `app.js` را به محتوای زیر جهت استفاده از میان افزار تغییر دهید

```
var http = require('http');
var express = require('express');
var app = express();
app.set('view engine', 'jade');
app.set('views', './views');
app.use(express.static('./public'));
// زمان پاسخ گوی را اضافه می کند
app.use(express.responseTime());
app.get('/', function(req, res) {
  res.render('index');
  [ 41 ]
});
```

<sup>11</sup> Parser

<sup>12</sup> Cache

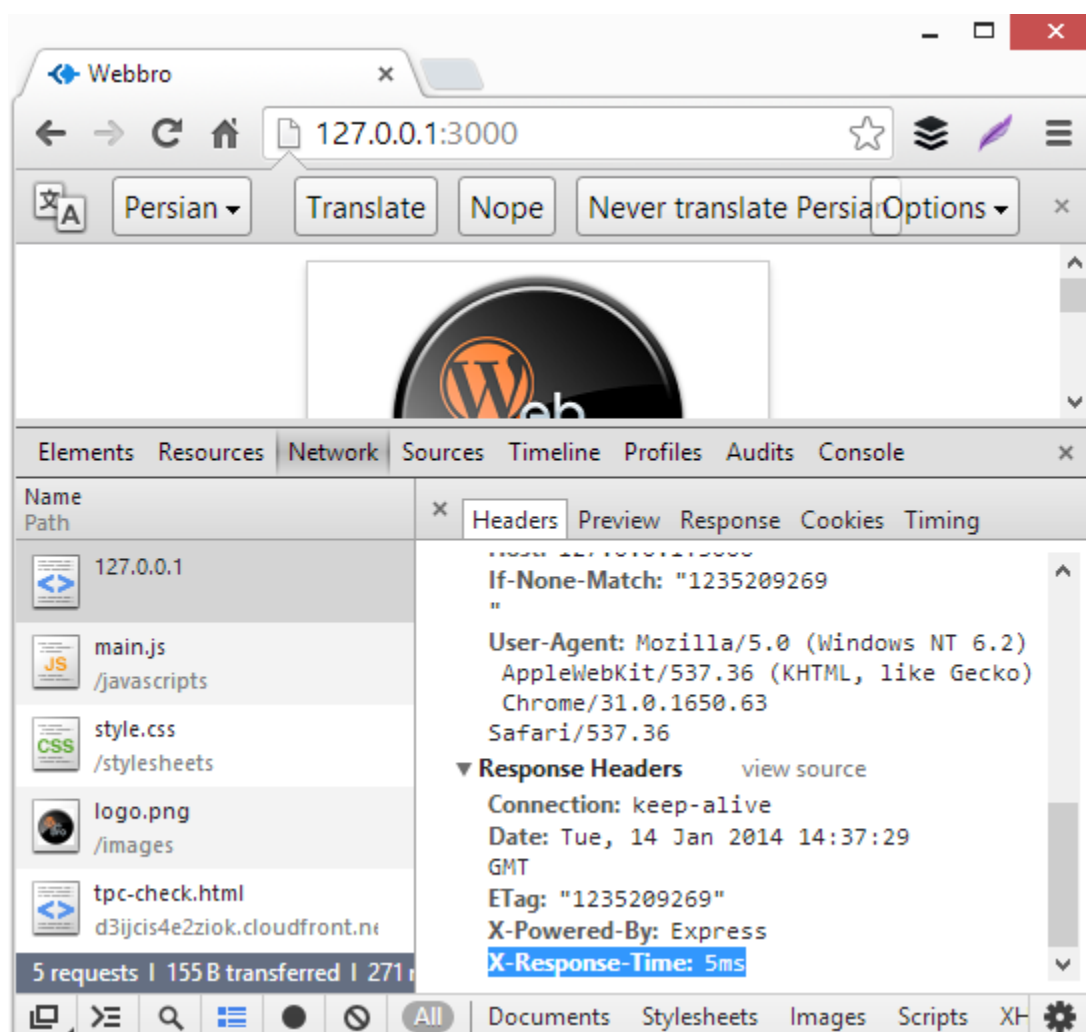
<sup>13</sup> [Vhost](#)

```
http.createServer(app).listen(3000, function() {  
  console.log('App started');  
});
```

اپ شماره ۳

خوب مثل همیشه با دستور `node app` اپ را اجرا کنید و بعد از اجرا خروجی تصویر زیر را مشاهده می کنید.

همانطور که در تصویر مشخص است از مرورگر کروم استفاده کرده ام که زمان پاسخگویی `http` به مرورگر در تب شبکه ابزار توسعه<sup>۱۴</sup> نمایش داده شده است.



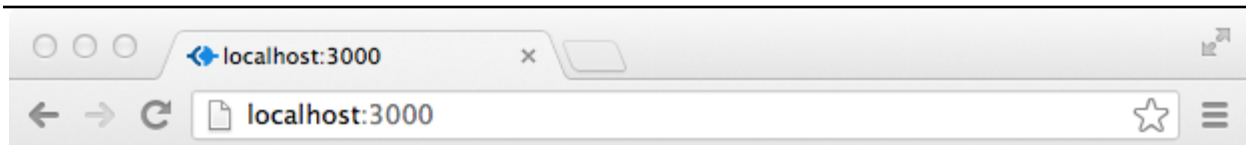
<sup>14</sup> [Chrome developer tool](#)

در مرحله بعدی می خواهیم میان افزار `handle error` را بررسی کنیم . برای بررسی این `handler` یک خطا تولید می کنیم و این عمل را با صدا زدن یک تابع تعریف نشده انجام می دهیم و بر طبق `errorHandler` توضیحات چگونگی این `Error` به صورت `HTML` نمایش داده خواهد شد به کد زیر توجه کنید .

```
var http = require('http');
var express = require('express');
var app = express();
app.set('view engine', 'jade');
app.set('views', './views');
app.use(express.static('./public'));
app.use(express.responseTime());
// اضافه کردن میان افزار هندلر ارور
app.use(express.errorHandler());
app.get('/', function(req, res) {
// تابع که تعریف نشده است جهت تولید خطا صدا می زنیم
fail();
});
http.createServer(app).listen(3000, function() {
console.log('App started');
});
```

کد بالا مانند به بقیه کد های `app.js` پیشین می باشد با این تفاوت که تابع `fail()` را فراخوانی کرده ایم بدون اینکه آن را تعریف کنیم جهت تولید خطا و دیدن چگونگی کارکرد `errorHandler()` که در این اپ همانطور که در کد بالا می بنید به اپ اضافه شده است .

مثل همیشه برای اجرای `app` از دستور `node app` در شاخه مورد نظر استفاده می کنید ( پروژه شماره ۴ ) و خروجی آن را در تصویر زیر می توانید مشاهده کنید



```
ReferenceError: fail is not defined
    at /Users/yaapa/projects/temp/app.js:15:3
    at callbacks
    (/Users/yaapa/projects/temp/node_modules/express/lib/router/index.js:162:37)
    at param
    (/Users/yaapa/projects/temp/node_modules/express/lib/router/index.js:136:11)
    at pass
    (/Users/yaapa/projects/temp/node_modules/express/lib/router/index.js:143:5)
    at Router._dispatch
    (/Users/yaapa/projects/temp/node_modules/express/lib/router/index.js:171:5)
    at Object.router
```

خروجی تصویر فوق مانند یک صفحه HTML نمی باشد در حقیقت شما می توانید تنظیم کنید که بصورت HTML نمایش داده شود. اما سوال چرا میان افزار errorHandler کار نکرد ؟

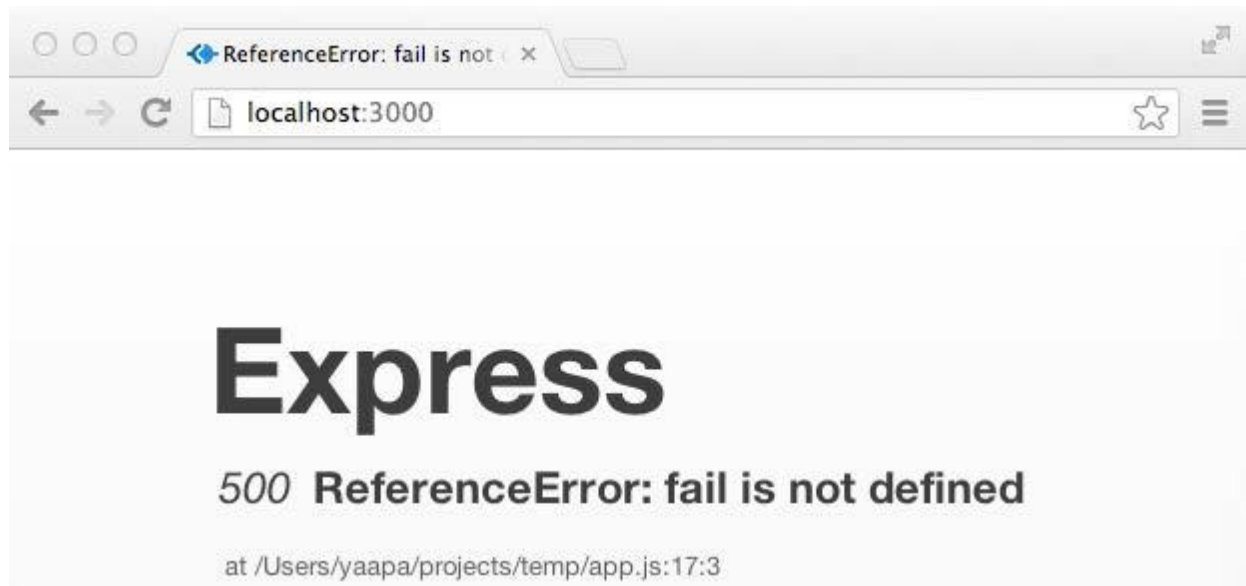
همانطور که در اول این مبحث اشاره شد بودن اضافه کردن میان افزار router بصورت مشخصی اپ مسیر تعریف شده را نمایش می دهد . مهمترین درخواست در میان افزار errorHandler این می باشد که پیش از آن باید میان افزار router را اضافه کنید . هیجان زده نشوید آنگونه که باید کار کند نمی شود .

طبق انتظارات و بدیهیات میان افزار router را جهت تولید و استفاده از errorHandler بکار می بندید که زیر را مشاهده کنید (پروژه شماره ۵)

```
var http = require('http');
var express = require('express');
var app = express();
app.set('view engine', 'jade');
app.set('views', './views');
app.use(express.static('./public'));
app.use(express.responseTime());
// اضافه کردن میان افزار روتر
app.use(app.router);
// اضافه کردن میان افزار ارورهندلر
app.use(express.errorHandler());
app.get('/', function(req, res) {
// صدا زدن تابع تعریف نشده جهت تولید خطا
```

```
fail();  
});  
http.createServer(app).listen(3000, function() {  
  console.log('App started');  
});
```

خوب کد بالا را اجرا کنید و خروجی مانند تصویر زیر مشاهده خواهید کرد



این بار `errorHandler` خروجی نمایش داد که انتظار آن می رفت در خروجی که بصورت HTML می باشد اعلام می کند تابع `fail` تعریف نشده است و پیام خطا مبدا با شماره ۵۰۰ را می دهد

شما می توانید از `errorHandler` هر زمان که نیاز داشتید از آن استفاده کنید. Express به صورت پیش فرض کار زیادی انجام نمی دهد در هنگام دی باگ اپ شما ولی این میان افزار بسیار مفید خواهد بود

اخیرا دیدم که شما میت وانید میان افزار خود را بنویسد البته اگر بخواهید. هر کاری که شما بخواهید انجام دهید با دو آبجکت (شی) `req` و `res` عملی می شود

---

<sup>15</sup> Object



## Express قدرت گرفته از ماژول Node

Express یک بسته با کتابخانه ی عظیم برای راه اندازی وب سایت نیست اما این به امر به این معنی نیست که قابلیت تولید تسک<sup>۱۶</sup> (وظایف) را ندارد

شما مجموعه ی عظیم از ماژول های نود را در ریجستری npm بصورت پلاگین در دسترس دارید و به آسانی می توانید از آن در اپ های خود استفاده کنید.

در فصل قبل توضیح دادیم ماژول نود را چگونه تولید و بنویسیم . ما از آن طریق می توانیم به نوشتن وسیع و قدرت مند در Express بپردازیم

شما با ماژول های نود می توانید خیلی چیزها را بنویسید و به سرانجام برسانید اما هر چیزی که سعی دارید بنویسید بصورت متن باز<sup>۱۷</sup> ماژول نود آماده استفاده می باشد و فقط احتیاج است شما ماژول درست را پیدا و نصب کنید و از آن در اپ خود استفاده کنید.

نکته npm بصورت عمومی در دسترسی همه توسعه دهندگان نود جهت انتشار و استفاده ماژول های نود موجود می باشد. ماژول های موجود با دستور npm نصب می شوند

لیست عظیم و بزرگی از ماژول های موجود در این آدرس موجود می باشد

<https://github.com/joyent/node/wiki>

چگونگی روش نصب NPM بوسیله یک فایل ini. بصورت مثال یک ماژول پارسر به نام iniparser را نصب و در اپ که می سازیم استفاده خواهیم کرد .

```
$ npm install iniparser
npm WARN package.json application-name@0.0.1 No README.md file
found!
npm http GET https://registry.npmjs.org/iniparser
npm http 304 https://registry.npmjs.org/iniparser
iniparser@1.0.5 node_modules/iniparser
```

---

<sup>16</sup> Task

<sup>17</sup> [Open Source : OS](#)

همانطور که در تکه کد بالا می بینید با دستور `npm install iniparser` در ترمینال سیستم عامل این ماژول نصب می شود .

حالا در ادامه ساخت اپ یک فایل به نام `config.ini` با محتوای زیر در مسیر اصلی اپ بسازید

اپ وب پرو `title =`

`port = 3000`

!شما عالی هستید `message =`

هنوز کار ما تمام نشده است حالا باید فایل `app.js` را بصورت زیر تغییر دهید .

```
var http = require('http');
var express = require('express');
var app = express();
// ماژول پارسر را اضافه میکند
var iniparser = require('iniparser');
// محتوای فایل کانفیگ را می خواند
var config = iniparser.parseSync('./config.ini');
// قسمت های زیر پیشتر توضیح داده شده است
app.set('view engine', 'jade');
app.set('views', './views');
app.use(express.static('./public'));
app.use(express.responseTime());
app.use(app.router);
app.use(express.errorHandler());
app.get('/', function(req, res) {
// دو متغیر کانفیگ به ویور پاس داده می شود
res.render('index', {title:config.title,
message:config.message});
});
http.createServer(app).listen(config.port, function() {
console.log('App started on port ' + config.port);
});
```

و فایل jade که وظیفه نمایش را دارد به شکل زیر تغییر می دهیم .

```
doctype html
html(lang="fa" dir="rtl")
title #{title}
script(src='javascripts/main.js')
link(rel='stylesheet', href='stylesheets/style.css')
body
  #content

    img(src='images/logo.png')
    p آمدید خوش
    P #{message}
    #smile
    a(href='http://www.webbro.ir', target='_blank')
Webbro
```

پروژه شماره ۶

حالا وقت اجراست . در تصویر زیر اجرای خروجی اپ فوق قرار دارد .



همانطور که متوجه شده اید می توانید با کمک `config.ini` از زبان فارسی برای دادن عنوان و ... غیر نیز استفاده کنید .پارسر از تمام استاندارد Unicode پشتیبانی می کند .

Express بصورت ذاتی توانایی پارس کردن فایل `.ini` را ندارد اما می توان با توسعه آن از طریق ماژول `iniparser` این قابلیت را به `express` اضافه کرد همانطور که در اپ این اتفاق افتاد.

### درخواست *log* به اپ

Express ماژولی دارد به نام `logger` که بسیار مفید برای ساخت و توسعه اپ ها می باشد . برای استفاده از این ماژول با تعریف خط زیر می توانید به راحتی آن را به اپ خود اضافه و از آن استفاده کنید

```
app.use(express.logger());
```

بدون هیچ تغییر ماژول `log` توضیحات ورودی را به همراه دارد . شما می توانید آن را از طریق نشانه های <sup>۱۸</sup> موجود که در زیر قرار دارد بصورت سفارشی از آن استفاده کنید .

نشانه	محتوا
<code>:req[header]</code>	درخواست بصورت ویژه با HTTP
<code>:res[header]</code>	پاسخ بصورت ویژه با حالت HTTP
<code>:http-version</code>	ورژن HTTP
<code>:response-time</code>	مدت زمان جهت تولید پاسخ
<code>:remote-addr</code>	آدرس IP کاربر
<code>:date</code>	تاریخ و زمان درخواست
<code>:method</code>	متد HTTP استفاده شده برای یک درخواست
<code>:url</code>	URL درخواست کننده
<code>:referrer</code>	URL مراجعه کننده URL جاری
<code>:user-agent</code>	اطلاعات کاربر
<code>:status</code>	وضعیت HTTP

---

<sup>18</sup> Token

و در زیر چگونگی فرمت استفاده از نشانه های بالا قرار دارد

```
app.use(express.logger({ format: ':remote-addr :method :url' }));
```

در زمان استفاده از خط بالا در اپ و بعد از مراجعه به صفحه در ترمینال خود پیام مشابه زیر را مشاهده خواهید کرد.

### 127.0.0.1 GET /

#### 127.0.0.1 GET /favicon.ico

بصورت پیش فرض خروجی logger در ترمینال نمایش داده می شود . اما ما می توانیم با کمک گزینه stream وضعیت log را در یک فایل ذخیره کنیم . کد زیر را مشاهده کنید .

```
var http = require('http');
var express = require('express');
var fs = require('fs');
var app = express();
app.use(express.logger({
  format: 'tiny',
  // ذخیره سازی لاگر در فایل
  stream: fs.createWriteStream('app.log', { 'flags': 'w' })
}));
...
```

Logger از چهار حالت تعریفی با فرمت های Tiny , Short , Default و Dev پشتیبانی می کند و شما می توانید بصورت خاص هر کدام را نیز از طریق زیر تعریف و استفاده کنید . به کد زیر توجه کنید :

```
app.use(express.logger('dev'));
```

### استفاده از یک فایل پیکربندی شده

بصورت عمومی نیاز به استفاده از فایل `ini` برای پیکربندی اپ های ما نمی باشد در مثال های بعدی متوجه چگونگی این موضوع خواهید شد . در مثال های که تاکنون دیدم فقط استفاده از ماژول `nod` را نشان دادیم نه توصیه های عملی .

حالا از زاویه ای چگونگی کار کرد `require()` را خواهیم دید. `nod` از پیکربندی فایل `JSON-based` بصورت پیش فرض پشتیبانی می کند . ایجاد یک فایل بوسیله یک آبجکت (شی) `JSON` و عمل ذخیره سازی آن بصورت فایل `json`. صورت می گیرد و برای لود فایل در اپ از `require()` استفاده می کنیم .

در زیر یک پیکربندی فایل `JSON-based` قرار دارد

```
{
  "development": {
    "db_host": "localhost",
    "db_user": "root",
    "db_pass": "root"
  },
  "production": {
    "db_host": "10.10.10.10",
    "db_user": "myappdb",
    "db_pass": "!p4ssw0rd#"
  }
}
```

کد فوق را در فایل `config.json` ذخیره کنید و خط زیر روش لود آن می باشد

---

```
var config = require('./config.json')[app.get('env')];
```

اپ خفن شما آمده اجرا است اما اگر می خواهید نتیجه خروجی را در ترمینال ببینید کد زیر را اضافه کنید .

```
console.log(config.db_host); // 192.168.1.9
console.log(config.db_user); // myappdb
console.log(config.db_pass); // !p4ssw0rd#
```

اپ شماره هفت

## Set و Get / پلکیشن

اپ Express یک تابع پیش تعریف شده به نام set دارد که پیشتر از این در اپ ها استفاده کرده ایم این تابع مقدارهای گوناگون داینامیکی را برای app از این طریق می توان مقدار دهی کرد . پیشتر از این دو نمونه زیر استفاده کرده ایم

```
app.set('view engine', 'jade');
app.set('views', './views');
```

مقدارهای از متغیرهای اپلکیشن را می توان بدست و استفاده کرد از طریق متد app.get() .

جدول زیر لیست از گزینه های که برای پیکر بندی اپ express می توانید از آنها استفاده کنید موجود می باشد

گزینه	معنی و منظور
env	وضعیت در حال اجرا . پیشنهاد می شود بصورت دستی مقدار دهی نشود (در بخش بعد بیشتر خواهید دانست )
trust proxy	فعال کردن پروکسی
json repacer	باز خواندن JSON
jsonp callback name	باز خوانی نام برای درخواست های JSONP
case sensitive routing	حساس بودن بزرگ و کوچک بودن نام route
strict routing	اسلش پایانی در آخر نام یک route
view cache	نمایش پشته
view engine	موتوری برای پروسس فایل ها
views	مسیری از فایل ها

## Environment متفاوت اکسپرس

در پروسه تولید نرم افزار سیستم ها توسعه داده می شود به شکلی که برای مراحل تست کاربر پذیری , مرحله تکمیلی و ارائه , تولید و برای ارائه جهت تولید نسخه نهایی . بصورت فنی به این زمینه از اجرای و نرم افزار را وضعیت <sup>۱۹</sup> می گویند .

توضیح بالا روش بسیار عمومی وضعیت نرم افزار است که در اکسپرس روش متفاوتی از روش معمول استفاده شده . برای مثال در وضعیت توسعه سازندگان اکسپرس تمایل دارند تا بنیند درباره جزئیات هر کدام از خطاهای نرم افزار را . بریم برای دیدن چگونگی کار کردن آن.

متد اکسپرس `app.get('env')` وضعیت جاری اپ را بر می گرداند . وضعیت یک اپ را نیز می توان تغییر داد اما پیشنهاد می شود این عمل بصورت دستی انجام نشود .

پیش از آن که تنظیمات پایه ای اپ را بصورت وضعیت انجام دهیم چگونگی کار کردن `app.get('env')` را توضیح می دهیم

زمانی که یک اپ اکسپرس شروع می شود به دنبال یک متغیر وضعیت می گردد و صدا می زند `NODE_ENV` از آبجکت `process.env` , اگر موفق به یافتن آن شد مقدار `NODE_ENV` علامتگذاری می شود به متغیر `env` اپ در صورت یافت نشدن مقدار `development` علامتگذاری می شود . متغیر اپ توسط `app.get` خوانده و توسط `app.set` نوشته می شود .

بصورت خلاصه : اگر `NODE_ENV` ماشین را تنظیم نکرده باشید وضعیت مقدار `development` خواهد گرفت . اگر قصد استفاده از یک ماشین توسعه را دارید نیاز به تنظیم کردن و هیچ مقدار دهی نمی باشد در غیر اینصورت بویژه برای تولید محصول درخواستی برای تنظیم اسم توسعه ارسال شود .

## پایان

---

<sup>19</sup> Environment



- 1- E-book : Concepts of Programming Languages 10<sup>th</sup>
- 2- E-book : Node beginner
- 3- Link[document]: <https://github.com/joyent/node>
- 4- Link [installation and API] <http://nodejs.org>
- 5- Link [about node.js] <http://css.dzone.com/articles/quick-introduction-how-nodejs>
- 6- <https://github.com/joyent/node/wiki/>